ADVANCING THE ART OF DIGITAL COMPOSITING

NUKE

FROM THE FOUNDRY

NUKE USER GUIDE
VERSION 5.1v5

Nuke™ USER GUIDE. Copyright © 2009 The Foundry Visionmongers Ltd. All Rights Reserved. Use of this User Guide and the Nuke software is subject to an End User License Agreement (the "EULA"), the terms of which are incorporated herein by reference. This User Guide and the Nuke software may be used or copied only in accordance with the terms of the EULA. This User Guide, the Nuke software and all intellectual property rights relating thereto are and shall remain the sole property of The Foundry Visionmongers Ltd. ("The Foundry") and/or The Foundry's licensors.

The EULA can be read in the Nuke User Guide, Appendix E.

The Foundry assumes no responsibility or liability for any errors or inaccuracies that may appear in this User Guide and this User Guide is subject to change without notice. The content of this User Guide is furnished for informational use only.

Except as permitted by the EULA, no part of this User Guide may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, recording or otherwise, without the prior written permission of The Foundry. To the extent that the EULA authorizes the making of copies of this User Guide, such copies shall be reproduced with all copyright, trademark and other proprietary rights notices included herein. The EULA expressly prohibits any action that could adversely affect the property rights of The Foundry and/or The Foundry's licensors, including, but not limited to, the removal of the following (or any other copyright, trademark or other proprietary rights notice included herein):

Nuke™ compositing software © 2009 The Foundry Visionmongers Ltd. All Rights Reserved.
Nuke™ is a trademark of The Foundry Visionmongers Ltd.

Digital Domain ® is a registered trademark of Digital Domain, Inc.

Primatte™ keyer tool © 1997-2009 Photron USA, Inc. All Rights Reserved.
Primatte™ is a trademark of IMAGICA Corp.
Primatte™ patent is held by IMAGICA Corp.

In addition to those names set forth on this page, the names of other actual companies and products mentioned in this User Guide (including, but not limited to, those set forth below) may be the trademarks or service marks, or registered trademarks or service marks, of their respective owners in the United States and/or other countries. No association with any company or product is intended or inferred by the mention of its name in this User Guide.

ACADEMY AWARD ® is a registered service mark of the Academy of Motion Picture Arts and Sciences.

Linux ® is a registered trademark of Linus Torvalds.

Windows ® is the registered trademark of Microsoft Corporation.

Mac, Mac OS, Tiger, Shake, Final Cut Pro and QuickTime are trademarks of Apple, Inc., registered in the U.S. and other countries.

Adobe ® and Photoshop ® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Maya ® is a registered trademark of Autodesk, Inc., in the USA and other countries.

Houdini ® is a registered trademark of Side Effects Software, Inc.

Boujou is a trademark of 2d3 Ltd.

3D-Equalizer is a trademark of Science.D.Visions.

Framecycler ® is a registered trademark of Iridas, Inc.

OpenGL ® is a trademark or registered trademark of Silicon Graphics, Inc., in the United States and/or other countries worldwide.

Cineon™ is a trademark of Eastman Kodak Company.

Stereoscopic images courtesy of Mr. Henry Chung, HKSC (http://www.stereoscopy.com/henry/). Images illustrating warping and morphing courtesy of Ron Brinkmann (http://www.digitalcompositing.com). Images from “The Day After Tomorrow” ©2004 courtesy of and copyright by 20th Century Fox. Images from "Stealth" courtesy of and copyright by Sony Pictures Inc. Images from "xXx" ©2002 courtesy of and copyright by Columbia Pictures Industries. All rights reserved by their respective owners in the United States and/or other countries.

The Foundry
1 Wardour Street
London
W1D 6PA

Rev: 16 April 2009

Contents

PREFACE 16
About this Manual 16
Getting Help 16
Viewing Online Help 16
Contacting Customer Support 17

GETTING STARTED 18
About the Chapters 18
Installation and Licensing 19
System Requirements 19
Windows and Linux 19
Mac OS X 19
Licensing Nuke 19
Obtaining Licenses 20
Installing Licenses 21
Further Reading 21
Installing Nuke 22
Installation on Windows 22
Installation on Linux 23
Installation on Mac OS X 24
Launching Nuke 25
Launching the Commercial Version 25
Launching the Nuke Personal Learning Edition (PLE) 27
About the Personal Learning Edition 27
Differences between the PLE and the Commercial Version of Nuke 27
Using the Interface 29
Understanding the Workflow 29
The Nuke Window 29
Panes and Panels 29
Tabbed Panels 31
Toolbar, Menu Bar and Content Menus 31
Using the Toolbar 33
Using the Menu Bar 35
Working with Nodes 35
Adding Nodes 35
Selecting Nodes 35
Renaming Nodes 37
Editing Nodes 37

Cloning Nodes 38
Disabling and Deleting Nodes 39
Connecting Nodes 39
Indicators on Nodes 41
Searching for Nodes 42
Viewing Information on Nodes 43
Customising the Node Display 43
Navigating Inside the Node Graph 44
Panning 44
Zooming 46
Fitting Selected Nodes in the Node Graph 46
Fitting the Node Tree in the Node Graph 46
Properties panels 46
Managing the Properties Bin 47
Controls That Appear on All Properties Panels 47
Displaying Parameters 49
Using Input Fields 50
Using Sliders 51
Separating Channels 51
Using the Colour Picker and Colour Controls 51
Using Colour Sliders and Colour Wheel 52
Animating Parameters 56
Working with Animated Parameters 56
Animated Parameters and the Curve Editor 57
Using the Curve Editor 57
Displaying Curves 57
Editing Curves 59
Viewers 66
Adding Viewer Nodes 66
Connecting Viewer Nodes 67
Toggling Views 67
Panning and Zooming the Viewer Window 68
Hiding Floating Viewers 68
Using the Viewer Controls 69
Using the Viewer Composite Display Modes 81
Using the File Browser 83
Undoing and Redoing 85
Customizing the Interface 86
Interface Layouts 86
Managing Scripts 90
Working with Multiple Image Formats 90
8-, 16-, and 32-Bit Image Processing 90
Setting Up Your Script 90

Name, Timespan, and Frame Rate 90
Full-size Formats 91
Proxy Mode 92
Caching 95
Saving Scripts and Recovering Backups 95
Saving Scripts 95
Automatic Backup of Scripts 96
Recovering Backups 97
Loading Image Sequences 97
Naming Conventions 99
Reformatting Image Sequences 99
Loading Nuke Scripts 100
Closing Nuke Scripts 100
Defining Frame Ranges 100
Filename Search and Replace 101
Displaying Script Information 102
Grouping Nodes in the Node Graph 102
Grouping Nodes with the Backdrop Node 102
Grouping Nodes with the Group Node 104
Adding Notes to the Node Graph 105

REFERENCE 107
Organisation of the Section 107

Reformatting Elements 109
Reformatting Images 109
Using the Reformat Node 109
Cropping Elements 113
Adjusting the Bounding Box 114
Resizing the Bounding Box 114
Copying a Bounding Box from One Input to Another 115
Adding a Black Outside Edge to the Bounding Box 116

Channels 117
Overview 117
Understanding Channels 117
Understanding Channel Sets (Layers) 117
Creating Channels and Channel Sets 117
Calling Channels 118
Selecting Input Channels 120
Selecting Masks 120
Tracing Channels 122
Renaming Channels 122

Removing Channels and Channel Sets 122
Swapping Channels 123
Channels from Input 1 123
Channels from Input 2 123
Channel Outputs 124
Assigning Constants 125
Creating Swap Channel Sets 125
In Summary 125
Merging Images 126
Layering Images Together with the Merge Node 126
Merge Operations 127
Generating Contact Sheets 133
Copying a Rectangle from One Image to Another 135
Colour Correction and Colour Space 138
Making Tonal Adjustments 138
Using Histograms 139
Sampling White and Black points 139
Making Basic Corrections 140
Using Sliders 141
Using Colour Curves 141
Making Hue, Saturation, and Value Adjustments 143
Correcting HSV 144
Correcting Hue 146
Correcting Saturation 147
Masking Colour Corrections 148
Applying Grain 149
Using Synthetic Grain 149
Using Practical Grain 150
Applying Mathematical Operations to Channels 153
Clamping Channel Values 153
Offsetting Channel Values 154
Inverting Channel Values 154
Multiplying Channel Values 155
Applying Expressions to Channel Values 155
Transforming the Colour Space 156
Overriding the Default Cineon Conversion 156
Making Other Colour Space Conversions 157
Changing the Viewer Colour Space 157
Transforming Elements 159
Transforming in 2D 159
Using the 2D Transformation Overlay 159
Choosing a Filtering Algorithm 160

Translating Elements 165
Rotating Elements 166
Scaling Elements 167
Skewing Elements 168
Applying Core Transformations in 2.5D 169
Adding a Card3D Node 169
Specifying the Order of Operations 169
Choosing a Filtering Algorithm 169
Using the 3D Transformation Handles 170
Translating Elements 171
Rotating Elements 172
Scaling Elements 172
Skewing Elements 172
Adding Motion Blur 173

Tracking and Stabilising 175

Tracking an Image 175
Activating Track Anchors 176
Positioning Track Anchors 177
Calculating the Track 178
Retracking Part of a Track 179
Editing Tracks 179
Manipulating the Track Overlays 179
Manipulating Track Curves and Smoothing Tracks 180
Tracking and Multiview Projects 181
Applying Tracking Data 182
Applying Tracking Data Using Tracker Controls 182
Applying Tracking Data Via Linking Expressions 183

Primatte 187

Accessing Primatte from Nuke 187
Primatte Basic Operation Tutorial 188
Auto-Compute 188
Select BG Colour 188
Clean BG Noise 190
Clean FG Noise 191
Spill Removal - Method #1 193
Spill Removal - Method #2 193
Spill Removal - Method #3 194
Repeatable Sampling Tools 194
The Spill Sampling Tools 195
The Matte Sampling Tools 195
The Detail Sampling Tools 196
Spill Replacement Options 196

No Suppression (no suppression): 196
Complemental Replacement Mode (complement): 197
Solid Colour Replacement Mode (solid colour): 197
Defocus Spill Replacement (defocused background): 198
Primatte Tools and Buttons 199
Initialize Section 199
Degrain Section 203
Actions Section 206
Fine Tuning Section 208
Spill Process Section 209
Output Section 210
The Primatte Algorithm 210
Explanation of how Primatte works 211
Explanation of how Primatte RT+ works 217
Explanation of how Primatte RT works 218
Contact Details 218
Main office 218
Primatte office 218
Proprietary Notices 219
Using Paint 220
Connecting the Paint Node 220
Applying Strokes 220
Using the Freehand Tool 221
Using the Reveal Tool 221
Using the Eraser Tool 222
Using the Clone Tool 223
Selecting Strokes for Editing 223
Editing Stroke Attributes 223
Editing Colour 224
Editing Opacity 224
Editing Brush Size 225
Editing Brush Hardness 226
Editing Comp Mode 226
Editing Stroke Timing 227
Editing Stroke Stack Order 227
Editing Stroke Vectors 228
Animating Paint Strokes 229
Copying, Pasting and Cutting Stroke Attributes 230
Copying Attributes 230
Pasting Attributes 230
Cutting Attributes 231
Copying, Pasting and Cutting Stroke Vectors 231
Copying Vectors 231

Pasting Vectors 231
Cutting Vectors 232
Paint and Stereoscopic Projects 232
Temporal Operations 233
Distorting Time 233
Simple Retiming 233
Interpolation 234
Frame-blending 235
OFlow Retiming 237
OFlow Parameters 238
Warping Clips 241
Global Frame Range and Speed 242
Applying the TimeBlur Filter 243
Editing Clips 244
Slipping Clips 244
Cutting Clips 244
Splicing Clips 245
Warping and Morphing Images 247
Warping 247
Warping Images Using the GridWarp Node 247
Warping an Image Using the SplineWarp Node 252
Animating Warps 255
Morphing 257
Creating Effects 261
Background Reflections on Foreground Elements 261
Creating Star Filter Effects on Image Highlights 263
Creating Text Overlays 265
Creating a Text Overlay 265
Repositioning and Transforming Text 268
Fonts 268
Changing the Text Colour 271
Analysing Frame Sequences 273
Analysing and Matching Frame Sequences 273
3D Compositing 277
Overview 277
Setting Up a Scene 278
The Scene Node 278
The ScanlineRender Node 278
The Camera Node 279
Using the 3D Viewer 280

3D Scene Geometry 282
Working with Cards 283
Working with Cubes 286
Working with Spheres 287
Working with OBJ Objects 288
Parenting to Axis Objects 288
Merging Objects 290
Object Material Properties 290
Projecting Textures onto Objects 294
Projecting Textures with the UVProject node 294
Projecting Textures with the Project3D Node 295
Merging Shaders 295
Object Display Properties 296
Transforming Objects 298
Using the Transform Handles 298
Transforming from the Node Properties Panel 299
Transformations and the Pivot Point 300
Using the TransformGeo Node 300
Modifying Object Shapes 302
Modifying Objects Using Lookup Curves 302
Modifying Objects Using a Power Function 304
Modifying Objects Using an Image 305
Modifying Objects Using a Perlin Noise Function 307
Modifying Objects Using a Distortion Function 308
Modifying Objects Using a Trilinear Interpolation 309
Lighting 309
Direct Light 310
Point Light 310
Spot Light 311
Environment Light 312
The Light Node 314
Manipulating Object Normals 315
Working with Cameras 315
Projection Cameras 316
First a Little Math.... 316
Setting Up the Projection Camera Script 317
Adding Motion Blur to the 3D Scene 319
Importing Channel Files, Cameras, Lights, Transforms, and Meshes from Other Applications 321
Applying Tracks to an Object 321
Working with .fbx Files 322
Importing Cameras from Boujou 326
Rendering a 3D Scene 327

Adjusting the Render Parameters 328
Working with Stereoscopic Projects 329
Setting Up Views for the Script 329
Loading Multi-View Images 331
Displaying Views in the Viewer 333
Selecting which Views to Apply Changes to 335
Splitting Views Off 335
Selecting the View to Process When Using the Paint Node 336
Performing Different Actions on Different Views 337
Reproducing Changes Made to One View 337
Reproducing Paint Strokes 338
Reproducing Roto Shapes 339
Reproducing X and Y Values 340
Swapping Views 341
Converting Images into Anaglyph 342
Changing Convergence 343
Rendering Stereoscopic Images 347
Previews and Rendering 349
Previewing Output 349
Previewing in a Nuke Viewer 349
Flipbooking Sequences 350
Rendering Output 351
Render Resolution and Format 351
Output (Write) Nodes 352
File Name Conventions for Rendered Images 355
Render Farms 356
Expressions 357
Understanding Expression Syntax 357
Linking Expressions 357
Adding Mathematical Functions to Expressions 359
Setting Interface Preferences 367
Displaying the Preferences Dialog 367
Changing Preferences 367
Saving Preferences 367
Resetting Preferences 368
The Available Preference Settings 368
Preferences Tab 369
Windows Tab 370
Appearance Tab 373
Node Colors Tab 374

Node Graph Tab 374
Viewers Tab 378
Script Editor Tab 379

The Script Editor and Python 381

Workflow 381
Using the Script Editor 381
Example Scripts 385
Creating Nodes and Setting Their Parameters 385
Assigning Variables 387
Adding Parameters to Node Controls 387
Connecting Nodes and Setting Their Inputs 388
Setting Default Values for Controls 389
Rendering with the Write Node 390
Listing a Node's Controls 390
Undoing and Redoing Actions 391
Frame Navigation 391
Overriding the Creation of a Particular Node 392
Getting Information on the Nuke Environment You Are Running 393
Clearing Out the Current Nuke (.nk) Script 394
Creating Views for a Stereoscopic Project 394
Adjusting Parameter Values in Stereoscopic Projects 394
Automating Procedures 394
Getting Help 395
More Documentation 395
Viewing More Examples 396
Using the Help Statement 396
Python on the Web 396
Using wxPython in Nuke to Write Custom GUIs 396
Installation 396
General Workflow 397
Using PyQt4 in Nuke to Write Custom GUIs 398
Installation 399
General Workflow 399
Example 399

Configuring Nuke 401

What Is a Terminal and How Do I Use One? 401
Command Line Operations 402
Environment Variables 406
Setting Environment Variables 406
Nuke Environment Variables 409
Loading Gizmos, Plug-ins, and TCL scripts 410
Loading Python Scripts 411

Loading OFX Plug-ins 411
Defining Common Favourite Directories 412
Handling File Paths Cross Platform 413
Defining Custom Menus and Toolbars 414
Defining Common Image Formats 419
Gizmos, Custom Plug-ins, and Generic TCL Scripts 419
Creating and Sourcing Gizmos 420
Custom Plug-ins 430
Sourcing TCL Procedure 430
Template Scripts 431
Defining Common Preferences 431
Altering a Script's Lookup Tables (LUTs) 433
Overview 433
Displaying, Adding, Editing, and Deleting LUTs 433
Selecting the LUT to Use 435
In the display_lut pulldown menu, select the LUT you want to use. 436
Default LUT settings 436
Example Cases 436

TUTORIALS 438
The Projects 438
Installing the Project Files 438

Tutorial 1: Compositing Basics 440
Starting Nuke 440
Using the Toolbar 441
Using the Menus 442
Customizing Your Layout 443
Saving Files and File Backup 444
Setting Up the Project 446
Working with Nodes 447
Connection Tips 451
Importing Image Sequences 452
Navigating Inside the Windows 455
Working with Viewers 456
Reformatting Images 459
Using Proxies and "Down-res" 460
Compositing Images 461
Colour-correcting Images 464
Masking Effects 465
To create and apply a bezier mask 465

Creating Flipbook Previews 466
Rendering Final Output 467
Epilogue 470

Tutorial 2: Tracking, Stabilising and Matchmoving 471
One-Point, Two-Point, Three-Point, Four 472
Open the Tutorial Project File 472
Tracking a Single Feature 473
Tracking Obscured Features 477
Stabilising Elements 478
Matchmoving Elements 481
Epilogue 484

Tutorial 3: Keying and Mattes 485
Open the Tutorial Project File 485
Keying with Primatte 486
Image-based Keying 492
Rotoscoping 498
Keying Video 504
Epilogue 512

Tutorial 4: 3D Integration 513
The Basic 3D System 513
The 3D Viewer 513
The Geometry or Scene Node 514
The Camera Node 514
The ScanlineRender Node 515
Open the Tutorial Project File 515
Setting Up a 3D System 516
Making a Scene 523
Merging and Constraining Objects 526
Animating a Scene 529
Working with Geometry 533
Lighting & Surface Properties 540
Epilogue 543

APPENDICES 545
Organisation of the Section 545

Appendix A: Hot Keys 546
Conventions 546
Node Graphs, Viewers, Curve Editors, Script Editors, and Properties Bins 546

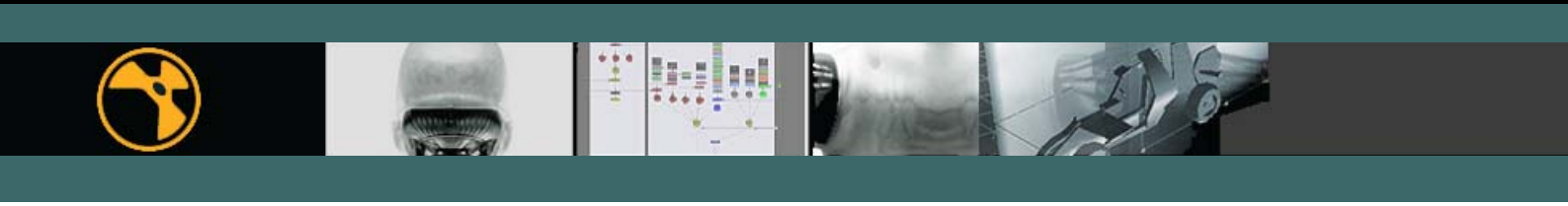
Properties panels 547
Node Graph 548
Editing 550
Viewers 550
3D Viewer 553
Bezier Draw 554
Curve Editor 554
Script Editor 555
Toolbar 556
Content Menus 556
Colour Picker 556

Appendix B: Supported File Formats 557
Supported Image Formats 557

Appendix C: Converting from Shake to Nuke 559
Terms (and Conditions) 559
Node Reference 561
Image Nodes 561
Color Nodes 562
Filter Nodes 564
Key Nodes 565
Layer Nodes 566
Transform Nodes 567
Warp Nodes 568
Other Nodes 569

Appendix D: Third Party Licenses 570

Appendix E: End User Licensing Agreement 579



PREFACE

Nuke is an Academy Award® winning compositor. It has been used to create extraordinary images on scores of feature films, including *Australia, The Dark Knight, Quantum of Solace, The Curious Case of Benjamin Button, The Golden Compass, Iron Man, Transformers*, *King Kong, Pirates of the Caribbean: At World's End*, and countless commercials and music videos.

About this Manual

This manual consists of four sections:

1. Getting Started, which teaches you how to acquire a licence and install Nuke, use the interface and work with script files.
2. Reference, which describes key features of Nuke in more detail. You can dip in and out of this section depending on what you're interested in.
3. Tutorials, which are designed to show you how to solve common compositing problems and help you really learn Nuke.
4. Appendices, which include the available hot keys, supported file formats, a Shake to Nuke conversion course, and the end user license agreement.

If you are new to Nuke, we recommend that you start by familiarising yourself with the Getting Started section and working your way through the Tutorials. These two sections should give you a good base to build on when creating your own scripts. If you then need an answer to a specific problem or want to learn how to use a specific feature, you can always turn to the Reference section. All the sections are colour-coded to make it easier for you to find what you are looking for.

Throughout this user guide, we assume you have a basic knowledge of computer graphics and digital compositing theory, as well as proficiency with the operating system for which Nuke is installed.

Note

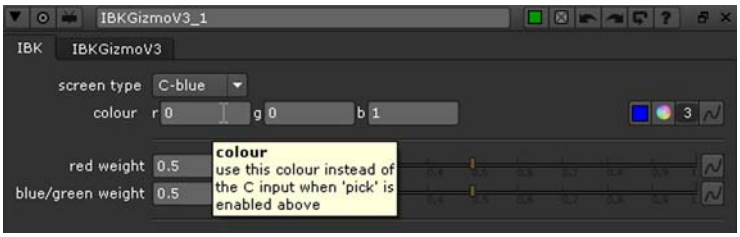
For the most up-to-date information, please see the Nuke product page and the latest Nuke user guide on our web site at www.thefoundry.co.uk.

Getting Help

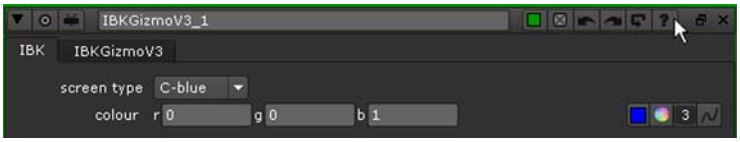
Viewing Online Help

Nuke features several forms of online help:

- Most controls offer concise instructions in the form of tool tips. To display the tool tips, move your mouse pointer over an interface control or a node parameter.



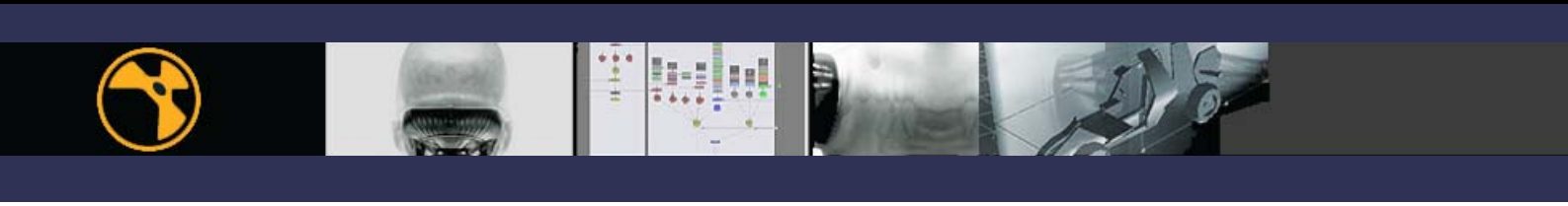
- Many properties panels include contextual descriptions of the node's parameters. To display these descriptions, click the **?** icon.



- Finally, you can click the **Help** menu to access the following:
  - **Key Assignments** - a list of hot keys.
  - **Documentation** - this user guide, the Nuke Developer Kit (NKD), and documentation for using FrameCycler, Python, TCL, and expressions in Nuke.
  - **Training** - FX PHD's Compositor's Guide to Nuke training videos, and a list of other training resources.
  - **Tutorials** - grain samples, and the files used with the tutorials in this user guide.
  - **Mailing Lists** - information on Nuke-related e-mail lists.

Contacting Customer Support

Should questions arise that this manual or the online help system fails to address, you can contact Customer Support directly via e-mail at **support@thefoundry.co.uk** or via telephone to our London office on **+44 (0)20 7434 0449** or to our Los Angeles office on (310) 399 4555 during office hours.



GETTING STARTED

By now, you are probably itching to dive into and play with Nuke, so let's get you started. This section provides you with all you need to start compositing with Nuke. For more detailed information on Nuke and its functions, see the *Reference* section on page 107 in this manual.

About the Chapters

Before you can start exploring Nuke's wonders, you of course need to install Nuke on your machine. For instructions on how to do so and launch either the commercial version of Nuke or the Nuke Personal Learning Edition, refer to Chapter 1: *Installation and Licensing* on page 19.

Once you have successfully installed and launched Nuke, you can sit back and start familiarising yourself with the interface. Chapter 2: *Using the Interface* on page 29 is designed to help you understand the workflow, the workspace and the different controls. It also provides you with information on adjusting the interface to suit your preferences.

Finally, to learn about scripts and script management, turn to Chapter 3: *Managing Scripts* on page 90.

1 Installation and Licensing

We know the installation and licensing of a new application can be a boring task that you just want to be done with as soon as possible. To help you with that, this chapter guides you to the point where you have an open workspace of Nuke in front of you and are ready to start compositing to your heart's content, whether it be with the commercial version of Nuke or the Nuke Personal Learning Edition (PLE).

System Requirements

Before you do anything else, check the system requirements to make sure your computer can run Nuke.

Windows and Linux

- 550 MHZ Pentium III or newer processor
- Windows XP (with Service Pack 2 or later), or CentOS 4.5.
- 5 GB disk space available for caching and temporary files
- 512 MB RAM (minimum requirement)
- Workstation-class graphics card, such as NVIDIA Quadro series, ATI FireGL series, or newer. Driver support for OpenGL 1.4.
  To enable optional GPU acceleration of certain effects, you need OpenGL 1.4 with support for floating point textures and GLSL.
- Display with at least 1280 x 1024 pixel resolution and 24-bit colour
- Three-button mouse

Note

To avoid graphical problems, such as text disappearing in the Viewer and Node Graph, it is important to keep your graphics card drivers up-to-date. Driver updates can be obtained from the web sites of the graphics card manufacturers (for example, www.nvidia.com and www.ati.com).

Mac OS X

- Intel and PPC. Mac OS X (10.5 Leopard)
- 5 GB of disk space available for caching and temporary files
- 512 MB of RAM (minimum requirement)
- AGP or PCI Express graphics card with at least 32 MB of video memory
- Display with at least 1280 x 1024 pixel resolution and 24-bit colour
- Three-button mouse

Licensing Nuke

If you simply want to try out or learn Nuke, you can run the Nuke Personal Learning Edition (PLE) without a license key. The PLE allows you to explore practically all Nuke's features, but prevents the commercial use of the application. To use the PLE, you only need to install Nuke on your machine (see *Installing Nuke* on page 22) and launch it in a special way described in

Launching the Nuke Personal Learning Edition (PLE) on page 27. *About the Personal Learning Edition* on page 27 also provides you with more information on the PLE and how it differs from the commercial version of Nuke.

To use the commercial version of Nuke, you need a valid license key. You can obtain one by purchasing Nuke or by requesting a time-limited demo license. The instructions below tell you how to get a license key and what to do with it. Without a valid license key, the commercial version of Nuke will fail to run.

Nuke uses FLEXlm encryption in the license keys. Node-locked (uncounted) and floating (counted) licenses are supported. We also supply a suite of tools to manage and monitor floating licenses running on a server across a network of machines. These tools are called Foundry FLEXlm Tools (FFT) and can be downloaded free of charge from our web site at www.thefoundry.co.uk/licensing. The Foundry FLEXlm Tools should be installed on the server.

Obtaining Licenses

You can request licenses by contacting The Foundry. To generate you a license key, we need to know your System ID. The System ID (sometimes called lmhostid) returns a unique number for your computer. We lock our license keys to the System ID.

To display your System ID, do any of the following:

- Download the Foundry System ID (FSID) utility from www.thefoundry.co.uk/licensing and run it. Your System ID will be displayed in the window that opens.
- Install and try to launch the commercial version Nuke without a valid license (see *Installing Nuke* on page 22 and *Launching Nuke* on page 25). Your System ID number is displayed in the license error you get.
- Download the Foundry FLEXlm Tools (FFT) free of charge from our web site and run the following command in a terminal shell:
  `./lmutil lmhostid`

Just so you know what a System ID number looks like, here's an example: 000ea641d7a1.

Once you have provided us with your System ID number and a license key has been generated for you, you will receive the license key in an e-mail or Internet download. The license key is contained in a text file called *foundry.lic*.

Here is an example node-locked (uncounted) license for Nuke that expires on 29 September 2009 for a computer running Nuke with a System ID of 000a957bfde5. Node-locked licenses allow you to run Nuke on one machine only.

```
INCREMENT nuke_i foundry 2008.0329 29-sep-2009 uncounted \
HOSTID=000a957bfde5 ISSUED=29-sep-2008 \
SIGN="00DA 99A9 E744 217E 8AD3 E7AF E289 C0C6 \
6B23 2891 AC01 0F50 E64D 8847 8B22 3A40 2BE9 \
A268 B7C2 4BC0 36AF"
INCREMENT nuke_r foundry 2008.0329 29-sep-2009 uncounted \
HOSTID=000a957bfde5 ISSUED=29-sep-2008 \
SIGN="03C9 1D0D 5503 EC34 2CAF 37C0 8731 5E57 \
```

```
06E8 C8CB E113 51EA 87C6 3BE8 242B 50AC 35EE \
6753 B3AB 3AC4 1559"
```

And here's an example of a permanent floating license for a server whose machine name is "red" with System ID 000ea641d7a1 communicating on port number 30001 that will enable any version of Nuke built before 31 July 2009 to run on up to 5 machines simultaneously.

```
SERVER red 000ea641d7a1 30001
VENDOR foundry
INCREMENT nuke_i foundry 2009.0731 permanent 5 ISSUED=15-jul-2009
SIGN="00DA 99A9 E744 217E 8AD3 E7AF
E289 C0C6 6B23 2891 AC01 0F50 E64D 8847 8B22
3A40 2BE9 A268 B7C2 4BC0 36AF"
INCREMENT nuke_r foundry 2009.0731 permanent 5 ISSUED=15-jul-2009
START=31-jul-2009 SIGN="03C9 1D0D 5503
EC34 2CAF 37C0 8731 5E57 06E8 C8CB E113 51EA
87C6 3BE8 242B 50AC 35EE 6753 B3AB 3AC4 1559"
```

For information on what to do with the foundry.lic file, see *Installing Licenses* below.

Installing Licenses

Once a license has been created for you, you will receive your license key (foundry.lic) in an e-mail or Internet download. You should also receive the Foundry License Installer (FLI) application to help you install the license key. The instructions below tell you what to do with these.

To install a license:

1. Make sure you have saved both the license key (foundry.lic) and the Foundry License Installer application in the same directory. You should receive these by e-mail or (if you purchased Nuke online) in a download.
2. Double-click on the Foundry License Installer application.
3. Click **Install**.

   This checks the license file and installs it into the correct directory.

   If you installed a node-locked license key (a license that allows you to run Nuke on one machine only), you're good to go. Proceed to *Launching Nuke* on page 25.

   If you are installing a floating license key, you will be asked whether you want to create a client license file. You should accept and save the file. For more information on how to install floating licenses, refer to the FFT user guide, which you can download from our web site.

Further Reading

There is a lot to learn about licenses, much of which is beyond the scope of this manual. For more information on licensing Nuke, displaying the System ID number, setting up a floating license server, adding new license keys and managing license usage across a network, you should read the Foundry FLEXlm Tools User Guide, which can be downloaded from our web site, www.thefoundry.co.uk/licensing.

Installing Nuke

Nuke 5.1 will install separately to any previous versions installed. Nuke 5.1 is available to download from our web site at www.thefoundry.co.uk. The downloads are in compressed tar format (tgz) for Linux, exe for Windows, and dmg format for Mac OS X. The installation procedure is the same for both the Nuke Personal Learning Edition and the commercial version of Nuke.

Installation on Windows

To install Nuke on Windows, do the following:

1. Download the correct installation file from our web site. Which installation file you download depends on whether you are using a 32- or a 64-bit machine.
2. Double-click on the file to install Nuke.
3. Follow the on-screen instructions. By default, Nuke is installed to *drive letter*:\Program Files\Nuke 5.1v5 (unless you're installing 32-bit Nuke on 64-bit Windows, in which case the default location is *drive letter*:\Program Files (x86)\Nuke 5.1v5).
4. Proceed to *Launching Nuke* on page 25.

Note

On Windows, if you install Nuke to a network drive to run from multiple computers, please ensure that the correct Microsoft run time libraries are installed on each machine that will run Nuke. To do this, run "vcredist_x86.exe" (32-bit) or "vcredist_x64.exe" (64-bit) on each machine. The appropriate one of these files can be found in the "VCRedist" subdirectory in the folder where Nuke is installed - for example, vcredist_x86.exe for the 32-bit version of Nuke.

Running Nuke without installing the libraries on your machine may work correctly, particularly as many systems (such as Windows Vista by default) will already have them. If the libraries are not present, Nuke will still run correctly, but some plug-ins may fail to load with error messages such as "This application has failed to start because the application configuration is incorrect. Reinstalling the application may fix this problem."

Please note that these libraries are set up automatically on the machine that runs the Nuke installer, so users installing on their local machine will not need to worry about this issue.

Installing Nuke from the Command Line

To install Nuke from the command line, do the following:

1. Download the correct .exe installation file from our web site at www.thefoundry.co.uk. Which installation file you download depends on whether you are using a 32- or a 64-bit machine.
2. To open a command prompt window, select **Start** > **All Programs** > **Accessories** > **Command Prompt**.
3. Use the **cd** (change directory) command to move to the directory where you saved the installation file. For example, if you saved the installation file in **C:\Temp**, use the following command and press **Return**:

   cd \Temp
4. To install Nuke, do one of the following:

- To install Nuke and display the installation dialog, type the name of the install file without the file extension and press **Return**:

  Nuke5.1v5-win-x86-release-32 (on a 32-bit machine)

  Nuke5.1v5-win-x86-release-64 (on a 64-bit machine)

- To install Nuke silently so that the installer does not prompt you for anything but displays a progress bar, enter **/silent** after the installation command:

  Nuke5.1v5-win-x86-release-32 /silent

  Nuke5.1v5-win-x86-release-64 /silent

- To install Nuke silently so that nothing is displayed, enter **/verysilent** after the installation command:

  Nuke5.1v5-win-x86-release-32 /verysilent

  Nuke5.1v5-win-x86-release-64 /verysilent

Note

By running a silent install of Nuke, you agree to the terms of the End User Licensing Agreement. To see this agreement, please refer to *Appendix E: End User Licensing Agreement* on page 579 or run the installer in standard non-silent mode.

Installation on Linux

To install Nuke on Linux, do the following:

1. Download the correct .tgz installation file from our web site at www.thefoundry.co.uk. Which installation file you download depends on whether you are using a 32- or a 64-bit machine.
2. Extract the installer from the tgz archive with the following terminal command:

   tar xvzf Nuke5.1v5-linux-x86-release-32.tgz (on a 32-bit machine)

   tar xvzf Nuke5.1v5-linux-x86-release-64.tgz (on a 64-bit machine)

   This will give you an installer file.
3. Run the installer.

   sudo ./Nuke5.1v5-linux-x86-release-32-installer (on a 32-bit machine)

   sudo ./Nuke5.1v5-linux-x86-release-64-installer (on a 64-bit machine)
4. Follow the on-screen instructions. By default, Nuke is installed to /usr/local/Nuke5.1v5.

5. If you didn't add a license key during the installation, do that now. Proceed to *Launching Nuke* on page 25.

Tip

To install Nuke silently, you can simply unzip the installer file. This creates the properly formed Nuke directory tree in the current directory.

By installing Nuke silently, you agree to the terms of the End User Licensing Agreement. To see this agreement, please refer to *Appendix E: End User Licensing Agreement* on page 579 or run the installer in standard non-silent mode.

Installing Nuke Remotely from the Command Line

If you need to install Nuke on render machines using the command line, do the following:

1. Download the correct .tgz installation file from our web site at www.thefoundry.co.uk. Which installation file you download depends on whether you are using a 32- or a 64-bit machine.
2. Extract the installer from the tgz archive with the following terminal command:

   tar xvzf Nuke5.1v5-linux-x86-release-32.tgz (on a 32-bit machine)

   tar xvzf Nuke5.1v5-linux-x86-release-64.tgz (on a 64-bit machine)

   This will give you an installer file.
3. Use the following terminal command to log in to your render machine as root:

   ssh root@render_machine

   Replace render_machine with the name of your render node.
4. Make a directory to install Nuke to:

   mkdir /usr/local/Nuke5.1v5
5. Copy the installer file from the machine that you downloaded it on to your render machine with a command like:

   scp root@download_machine:/tmp/Nuke5.1v5-linux-x86-release-32-installer root@render_machine:/usr/local/Nuke5.1v5/ (on a 32-bit machine)

   scp root@download_machine:/tmp/Nuke5.1v5-linux-x86-release-64-installer root@render_machine:/usr/local/Nuke5.1v5/ (on a 64-bit machine)

   Replace *download_machine* with the name of the machine you downloaded the installer file to, and *render_machine* with the name of your render node.
6. Unzip the installer file to unpack its contents into your Nuke directory:

   cd /usr/local/Nuke5.1v5unzip Nuke5.1v5-linux-x86-release-32-installer (on a 32-bit machine)

   unzip Nuke5.1v5-linux-x86-release-64-installer (on a 64-bit machine)
7. Repeat steps 3-6 for each render machine.

Installation on Mac OS X

To install Nuke on Mac OS X, do the following:

1. Download the .dmg installation file from our web site at www.thefoundry.co.uk.

2. Double-click on the dmg archive to extract the pkg installer. A pkg file is created.
3. Double-click on the pkg file.
4. Follow the on-screen instructions to install Nuke. By default, Nuke is installed to /Applications/Nuke5.1v5.
5. Proceed to *Launching Nuke* on page 25.

Installing Nuke Silently from the Command Line

1. Download the .dmg installation file from our web site at www.thefoundry.co.uk.
2. Launch a Terminal window.
3. To mount the dmg installation file, use the **hdiutil attach** command with the directory where you saved the installation file. For example, if you saved the installation file in **Builds/Nuke**, use the following command:

   hdiutil attach /Builds/Nuke/Nuke5.1v5-mac-x86ppc-universal-release-32.dmg
4. Enter the following command:

   pushd /Volumes/Nuke5.1v5/

   This stores the directory path in memory, so it can be returned to later.
5. To install Nuke, use the following command:

   sudo installer -pkg Nuke5.1v5-mac-x86ppc-universal-release-32.pkg -target "/"

   You are prompted for a password.
6. Enter the following command:

   popd

   This changes to the directory stored by the pushd command.
7. Finally, use the following command to eject the mounted disk image:

   hdiutil detach /Volumes/Nuke5.1v5

Note

By running a silent install of Nuke, you agree to the terms of the End User Licensing Agreement. To see this agreement, please refer to *Appendix E: End User Licensing Agreement* on page 579 or run the installer in standard non-silent mode.

Launching Nuke

With the installation and licensing out of the way, you're ready to start compositing with Nuke. Depending on whether you want to use the Nuke Personal Learning Edition or the commercial version of Nuke, the procedure for launching Nuke is slightly different. Both are described below, beginning with the procedure for the commercial version.

Launching the Commercial Version

To launch the commercial version of Nuke, do one of the following:

On Windows

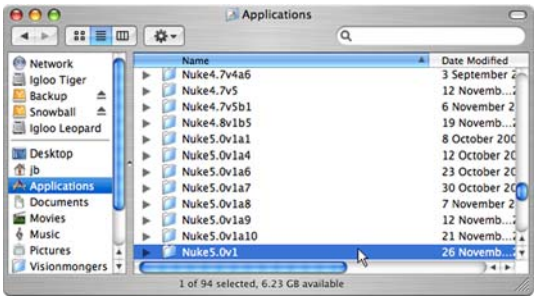
- Double-click the Nuke icon on the Desktop.
- Select **Nuke5.1v5 (32/64 bit)** from **Start** > **All Programs** > **The Foundry > Nuke5.1v5 (32/64 bit)**.

- Using a command prompt, navigate to the Nuke application directory (by default, **\Program Files\Nuke5.1v5** or \Program Files (x86)\Nuke 5.1v5), and enter **nuke5.1**.

The Nuke graphical interface appears.

On Mac OS X

- Click the Nuke dock icon.
- Open the Nuke application directory (by default, **/Applications/Nuke5.1v5/**), and double-click the **Nuke** icon (or list item).

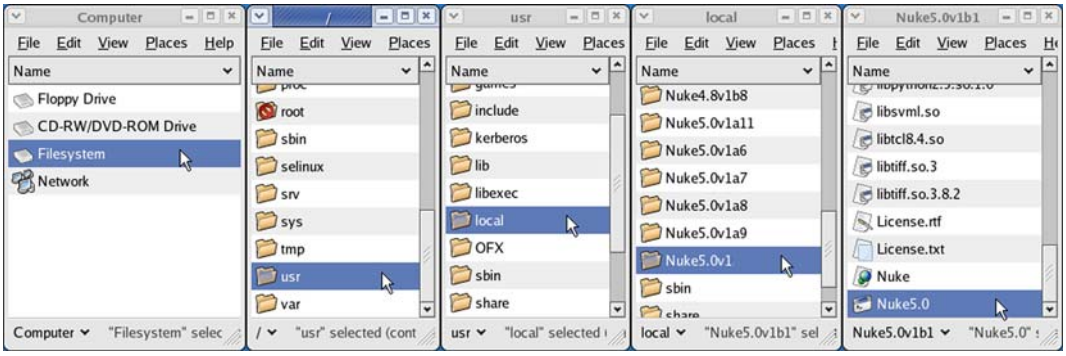


- Open a terminal, navigate to **/Applications/Nuke5.1v5/Nuke5.1v5.app/Contents/MacOS**, and enter **./Nuke5.1**.

The Nuke graphical interface appears.

On Linux

- Double-click the Nuke icon on the Desktop.
- Open the Nuke application directory (by default, **/usr/local/Nuke5.1v5/**) and double-click the **Nuke** icon (or list item).



- Open a terminal, navigate to the Nuke application directory (by default, **/usr/local/Nuke**), and enter **./Nuke5.1**.

The Nuke graphical interface appears.

Launching the Nuke Personal Learning Edition (PLE)

You launch the PLE from a terminal (a window where you can enter commands directly rather than making selections through a user interface).

To launch the PLE:

- On Windows: Select the Personal learning Edition from **Start** > **All Programs** > **The Foundry** > **Nuke5.1v5.**
  OR
  Using a command prompt, navigate to the Nuke application directory (by default, /Program Files/Nuke5.1v5), and enter **nuke5.1 --ple**.
- On Linux: Open a terminal, navigate to the Nuke application directory (by default, /usr/local/Nuke), and enter **./Nuke5.1 --ple**.
- On Mac OS X: Open a terminal, navigate to /Applications/Nuke5.1v5/Nuke5.1v5.app/Contents/MacOS, and enter **./Nuke5.1 --ple**.

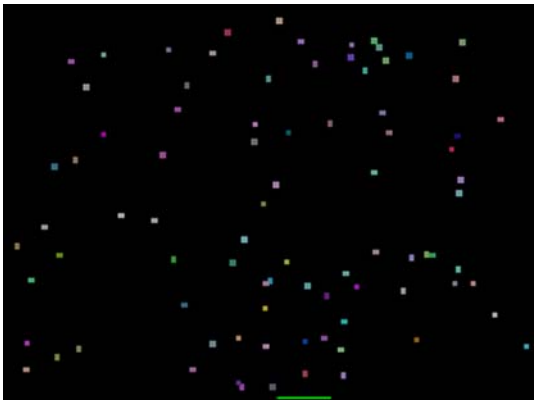
About the Personal Learning Edition

The Nuke Personal Learning Edition is a special version of Nuke that you can run without a license. The Personal Learning Edition is meant for personal, educational, and other non-commercial use. It is aimed at students, industry professionals, and others interested in Nuke. It includes all the features of the commercial version of Nuke, offering you a chance to explore and learn the application fully while using it from the comfort of your own home.

Differences between the PLE and the Commercial Version of Nuke

The PLE is a fully functional version of Nuke, but, being aimed for non-commercial use only, it does differ from the commercial version in some aspects. Here are the main differences:

- **Watermark**. The PLE displays a watermark (shown below) on any images in the Viewer as well as images rendered out to files. This is to prevent the commercial use of the images.



- **External data storage**. All external data storage is encrypted in the PLE, including Nuke scripts (these are saved with the extension .nkple), gizmos (saved with the extension .gzple), and copying to the clipboard. Among other things, this means the PLE saves files in an encrypted format, unlike the commercial version of Nuke, which saves scripts unencrypted as plain text. The commercial version of Nuke cannot load files created with the PLE.
  The PLE, however, can load scripts and gizmos created with the commercial version.
- **Scripting**. In PLE mode, Nuke restricts the amount of nodes that can be retrieved at a time by scripting. Functions such as "nuke.allNodes()" in Python will return only the first 10 nodes available rather than all of them at once, and scripts written to iterate through the Node Graph will not be able to retrieve any more nodes beyond a set point. The commercial version of Nuke can retrieve any and all nodes at any time as the command names would suggest.
- **WriteGeo**. The WriteGeo node is disabled in the PLE.
- **Primatte**. The Primatte node is disabled in the PLE.
- **FrameCycler**. FrameCycler is disabled in the PLE.
- **Plug-ins**. Only plug-ins that are shipped with Nuke, such as OFlow, can be used in the PLE. OFX plug-ins (such as The Foundry's Furnace and Tinder) and custom plug-ins compiled with the NDK can only be used in the commercial version of Nuke.

In other respects, the PLE contains all the functionality of the commercial version of Nuke.

2 Using the Interface

This chapter is designed to help you understand Nuke's workflow, learn how to use the interface, and customise the interface to suit your preferences.

Understanding the Workflow

Nuke utilizes a node-based work flow, where you connect a series of nodes to read, process, and manipulate images. Each node in the project—an image keyer, a colour-correction, or a blur filter, for example—performs an operation and contributes to the output.

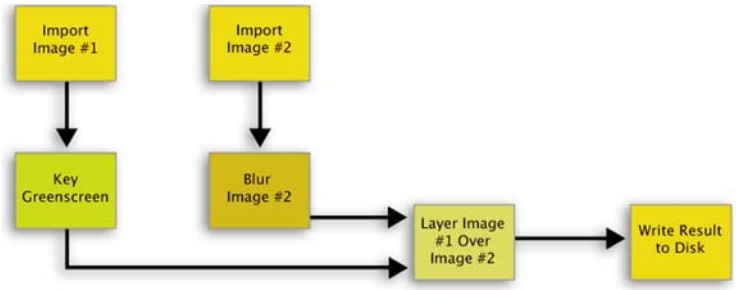


Figure 2-1: A Nuke project consists of a network of linked operators called *nodes*.

Saved projects are called *script files*. You can open a Nuke project file in a text editor, and you will see a series of sequential commands which are interpreted and executed when you render the output.

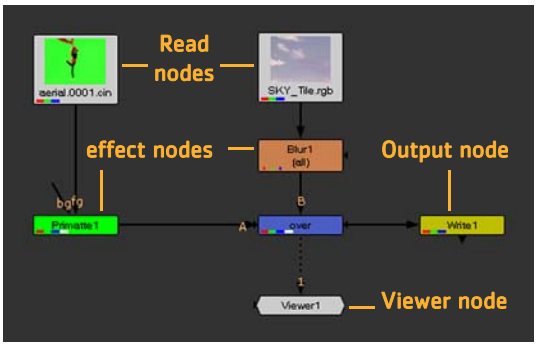


Figure 2-2: A simple Nuke script.

In Figure 2-2, you see an example of a very simple Nuke script. Two *Read* nodes reference image sequences on disk. Effect nodes extract a matte and blur an image. A merge node (named *over*) composites the foreground image over the background. Finally, a *Write* node renders and outputs the completed composite to disk. You'll also see a *Viewer* node, which displays the output of any node in the script.

The Nuke Window

Panes and Panels

Nuke's main window is divided into three panes: the Node Graph/Curve Editor pane, the Properties/Script Editor pane, and the Viewer pane.

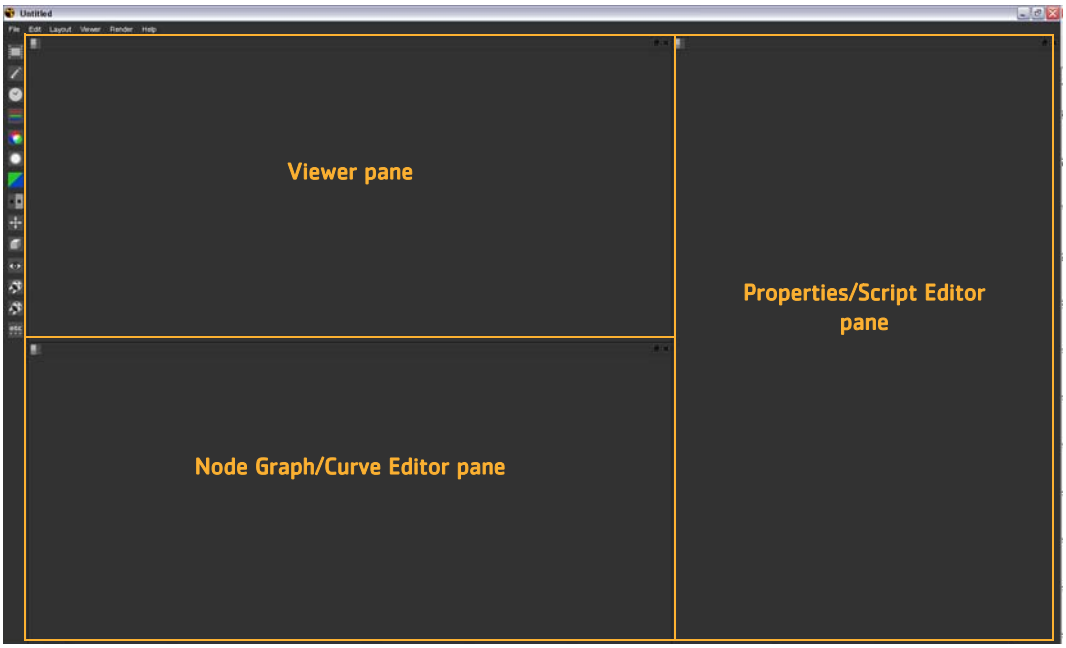


Figure 2-3: Empty main window.

Onto these panes, you can add the following panels:

- Toolbars for selecting nodes
- Node Graphs (also known as DAGs) for building node trees
- Curve Editors for editing animation curves
- Properties Bins for adjusting the nodes' controls
- Viewers for previewing the output
- Script Editors for executing Python commands.

By default, there is a Node Graph panel in the lower left corner, a Viewer panel in the top left corner, and a Properties Bin on the right, as shown in Figure 2-6.

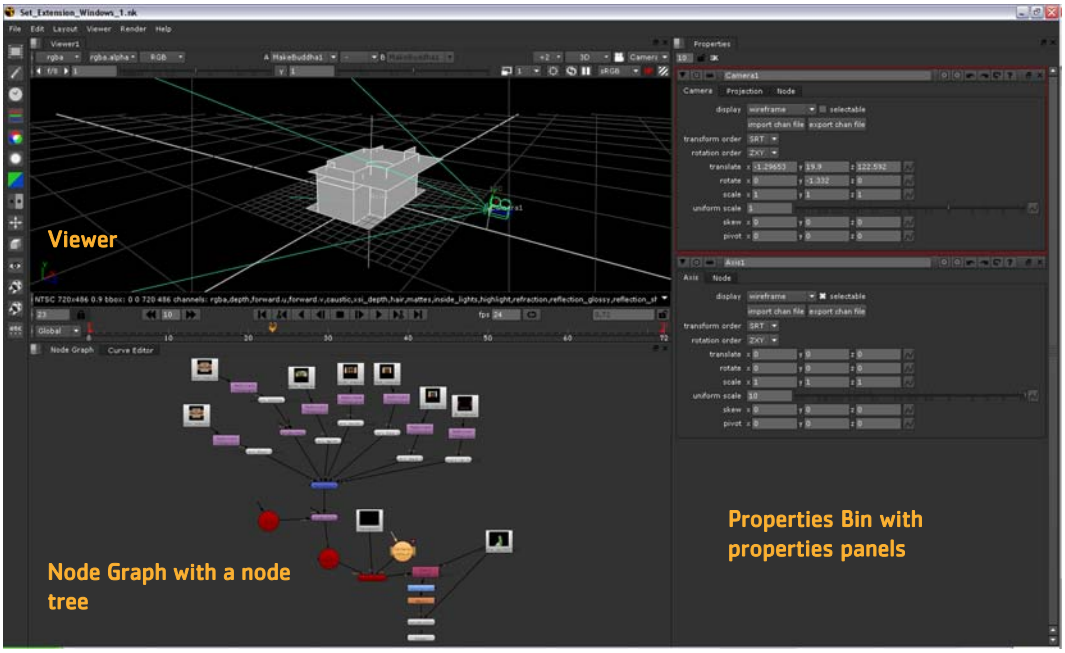


Figure 2-4: Main window with a Node Graph, Viewer, and Properties Bin.

The Node Graph is where you add nodes and build your node tree. When you add a node to the panel, its properties panel appears in the Properties Bin on the right. This is where you can adjust the node to produce the effect you're after. To check the result, you can view the output in a Viewer.

You can open more panels using the content menus described in *Toolbar, Menu Bar and Content Menus* below. You can add several panels on the same pane and switch between them by using the tabbed pages on top of the pane.

Tabbed Panels

Panes are divided into tabbed panels on the top of the pane. To go to a different tab, simply click on the tab name.

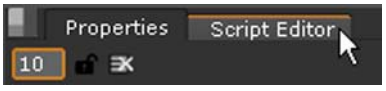


Figure 2-5: Moving from the Properties tab to the Script Editor tab.

Toolbar, Menu Bar and Content Menus

The toolbar is located on the left-hand side of the Nuke window. By default, it consists of twelve icons. The different nodes are grouped under these icons based on their functions. You use the toolbar to add nodes to the Node Graph.

The menu bar is located on top of the Nuke window. Its menus, such as the **File** or **Edit** menu, let you perform more general actions related to the whole script, the Viewers, or editing, rather than certain individual nodes.

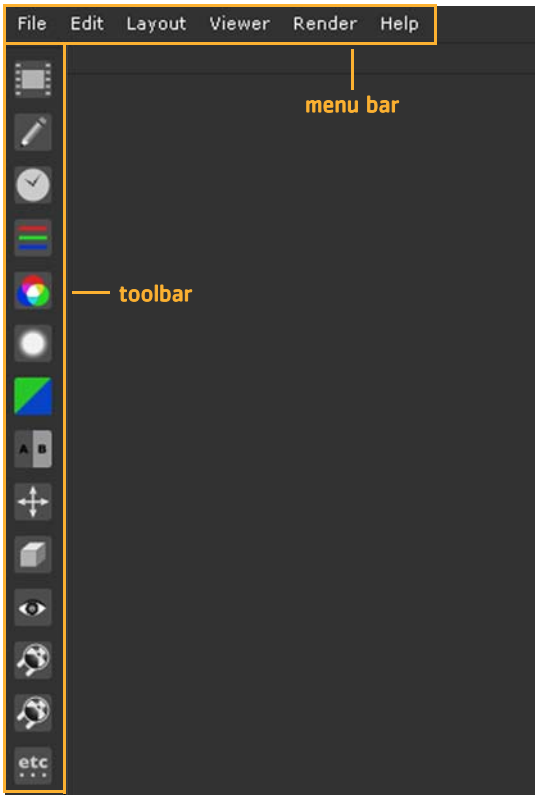


Figure 2-6: The toolbar and the menu bar.

In addition to the toolbar and the menu bar, you should also familiarise yourself with the content menus. They are the gray checkered boxes in the top left corner of each pane. If you click on the box, a pop-up menu opens. You can use the options in the pop-up menu to customise the window layout.

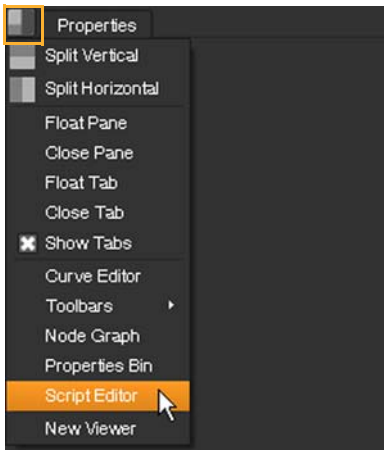


Figure 2-7: Using the content menus.

Finally, to work faster, you can right-click on the different panels to display a pop-up menu with options related to that particular panel.

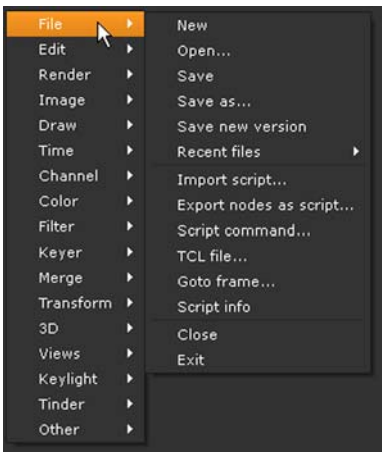


Figure 2-8: A right-click menu in the Node Graph.

Using the Toolbar

Nuke's toolbar includes the following icons:

Icon	Functions
Image	Image read and write nodes, built-in Nuke elements, and Viewer nodes.
Draw	Roto shapes, paint tools, film grain, fills, lens flares, sparkles, other vector-based image tools.
Time	Retiming image sequences.
Channel	Channel management.
Color	Applying colour correction effects.
Filter	Applying convolve filters, such as blur, sharpen, edge detect, and erode.
Keyer	Extracting procedural mattes.
Merge	Layering background and foreground elements.
Transform	Translating, scaling, tracking, and stabilizing elements.

Icon	Functions
3D	3D compositing nodes and tools.
Views	Nodes for working with views and stereoscopic or multi-view material.
	Any installed plug-ins and custom menus that do not have their own icon.
Other	Additional operators for script and Viewer management.

To display a tool tip that explains the icon's function, move your mouse pointer over the icon.

To make selections from the toolbar, click on an icon and select an option from the menu that appears.

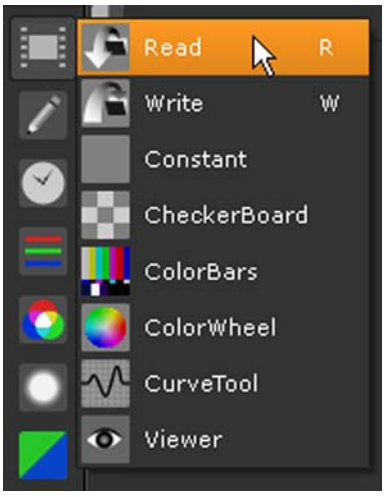


Figure 2-9: Selecting a node from the toolbar.

To quickly browse through the menus in the toolbar, click and drag over the icons. Nuke opens and closes the menus as you drag over them, making it easy to search for a particular node or find out what the available menus contain.

Tip

You can press the middle mouse button on a menu icon to repeat the last item used from that menu. For example, if you first select a Blur node from the **Filter** menu, you can then add another Blur node by simply pressing the middle mouse button on the **Filter** icon.

Using the Menu Bar

The Nuke menu bar includes these functions:

Menu	Functions
File	Commands for disk operations, including loading, saving, and importing scripts.
Edit	Editing functions, preferences, and project settings.
Layout	Restoring and saving layouts.
Viewer	Adding and connecting Viewers.
Render	Rendering the output.
Help	Accessing a list of hot keys, user documentation, training resources, tutorial files, and Nuke-related e-mail lists.

To quickly browse through the available menus and see what they contain, click on a menu and move the mouse pointer over other menus. Nuke opens and closes the menus as you go.

Working with Nodes

Nodes are the basic building blocks of any composite. To create a new compositing script, you insert and connect nodes to form a network of the operations you want to perform to layer and manipulate your images.

Adding Nodes

You add nodes using the toolbar. When you add a node, Nuke automatically connects it to the last selected node.

To add a node:

1. To select the existing node that you want the new node to follow, click on the node once.
2. Click an icon on the toolbar and select a node from the menu that appears. For example, if you want to add a Blur node, click the **Filter** icon and select **Blur**.

Tip

You can also add nodes by pressing the **Tab** key on the Node Graph and starting to type the name of the node. This opens a prompt displaying a list of matches. Select the node you want to add from the list either by clicking on it or scrolling to it with the **Up** and **Down** arrows and pressing **Enter**. To add the last node you created using this method, simply press **Tab** and then **Enter**.

Alternatively, you can add nodes using the right-click menu on the Node Graph or keyboard shortcuts (most menus in the toolbar include a note of the relevant hot key next to the item in question).

Selecting Nodes

Nuke offers a number of options for selecting nodes. Selected nodes display in a highlight colour defined in your preferences. The default highlight colour is light yellow.

To select a single node:
Click once on the node.

To select multiple nodes:
Press **Shift** while clicking on each node you want to select.
OR
Drag on the workspace to draw a marquee. Nuke selects all nodes inscribed by the marquee.

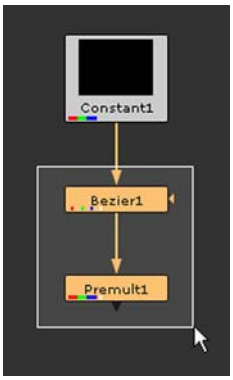


Figure 2-10: Selecting multiple nodes with a marquee.

To select all upstream nodes:
Press **Ctrl** (Mac users press **Cmd**) while dragging on a node. Nuke selects all nodes that feed data to the selected node.
Or press **Ctrl/Cmd + click** to highlight these upsteam nodes.
Or press **Ctrl/Cmd + Shift + click** to select these upsteam nodes.

To select all nodes in a script:
Select **Edit** > **Select all** (or press **Ctrl/Cmd**+**A**).

To select nodes by name:
1. Choose **Edit** > **Search**, or press the forward slash (/).
   A dialog appears.
2. Type an alphanumeric string that is included in the names of the nodes you wish to select. Click **OK**.

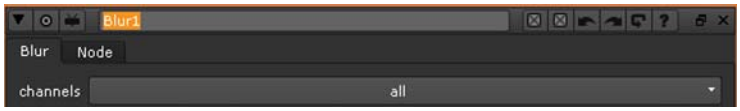
Tip

When typing the above alphanumeric search string, you can use asterisks (*) and question marks (?) as wild cards. An asterisk stands for multiple alphanumeric characters. A question mark represents just one character.

Renaming Nodes

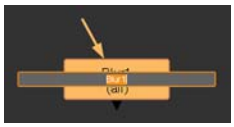
To rename a node:

1. Double-click on the node to open its properties panel.
2. In the title field on top of the properties panel, you should see the current name of the node. Delete that name and enter a new name in its place.



OR

1. Click on the node in the Node Graph to select it.
2. Press N.
3. Enter a new name for the node in the rename field that appears on top of the node.



Editing Nodes

To copy, paste, and perform other editing functions in the compositing tree, you can use the standard editing keys (for example, **Ctrl/Cmd+C** to copy, and **Ctrl/Cmd+V** to paste). You can copy nodes to files or memory. Copied nodes inherit the values of their parent, but these values, unlike those in cloned nodes (see below), are not actively linked—that is, you can assign different values to the original and the copy.

When you paste nodes, Nuke automatically connects them to the node that is selected before the paste operation. If you don't want to connect anything, click on a blank area of the workspace to deselect any selected nodes before pasting.

To copy nodes to memory:

1. Select the node or nodes you want to copy.
2. Choose **Edit** > **Copy** (or press **Ctrl/Cmd+C**).

To copy nodes to files:

1. Select the node or nodes you want to copy.
2. Choose **File** > **Export nodes as script**.
3. Navigate to the directory where you want to store the node as a file.
4. Type a name for the node(s) at the end of the pathway, followed by the extension *.nk*.

To cut nodes:

1. Select the node or nodes you want to cut.

2. Choose **Edit** > **Cut** (or press **Ctrl/Cmd+X**).
   Nuke removes the node(s) from the script and writes the node(s) to memory.

To paste nodes from memory:

1. Select the node that you want the pasted node to follow.
2. Choose **Edit** > **Paste** (or press **Ctrl/Cmd+V**).
   Nuke adds the nodes to the script, connecting them to the node you selected in step 1.

To paste nodes from files:

1. Select the node that you want the pasted node to follow.
2. Choose **File** > **Import script**.
3. Navigate to the directory that stores the node file.
4. Select the node file, and click **Import**.
   Nuke adds the nodes described by the file to the node you selected in step 1.

Cloning Nodes

You can clone nodes in preparation for pasting them elsewhere in a script. Cloned nodes inherit the values of their parent, but unlike copied nodes, they also maintain an active link with their parents' values. If you alter the values of one, the other automatically inherits these changes.

Clones are helpful for maintaining consistent setups across multiple elements. For example, you might use clones to apply an identical film grain setup to a series of elements shot on the same stock. Should you need later to make changes to the setup, these changes would automatically ripple throughout the script.

Note

You cannot clone gizmos. This applies to both gizmos created in your facility and the nodes in the Nuke default Toolbar that are in fact gizmos, such as the LightWrap node, the Grain node, and the IBK nodes.

For more information on gizmos, see *Gizmos, Custom Plug-ins, and Generic TCL Scripts* on page 419.

To clone nodes:

1. Select the node or nodes you want to clone.
2. Choose **Edit** > **Clone** or (press **Alt+K**).
   Nuke clones the node(s), whilst maintaining an active link to the parental node(s). The clone status is indicated with an orange line that connects the clone to its parent node. The nodes also share the same name.

To declone nodes:

1. Select the node or nodes you want to declone.
2. Choose **Edit** > **Declone** (or press **Alt+Shift+K**).

Nuke removes the clone status of the selected nodes.

Disabling and Deleting Nodes

To disable nodes:

1. Select the node or nodes you want to disable.
2. Select **Edit** > **Node** > **Disable\Enable** (or press **D**).
   Nuke cancels the node(s)'s effect on the data stream.

To re-enable nodes:

1. Select the node or nodes you want to re-enable.
2. Select **Edit** > **Node** > **Disable\Enable** (or press **D**).

To delete nodes:

1. Select the node or nodes you want to delete.
2. Select **Edit** > **Erase** (or press **Delete**).
   Nuke removes the node(s) from the script.

Connecting Nodes

As discussed, when you add or paste nodes into to a script, Nuke automatically generates pipes between the currently selected node and the new nodes. As you build up a script, you'll need to move these pipes, or run new pipes between nodes. In Nuke, you make such modifications by dragging on the back end of the pipe (the end without the arrowhead).

To disconnect a node:

Drag the head or tail of the connecting arrow to an empty area of the workspace.
OR
Select the lower node in the tree and press **Ctrl+D** (Mac users press **Cmd+D**).

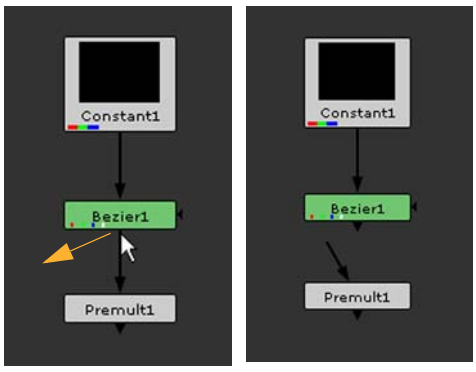


Figure 2-11: Disconnecting a pipe.

To reconnect a node:

Drag on the head or tail of the connecting arrow and drop it over the centre of the node to which you want to connect.

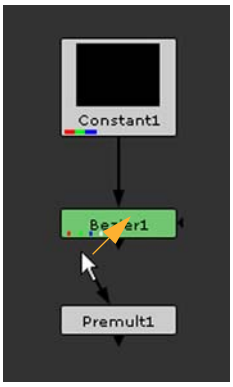


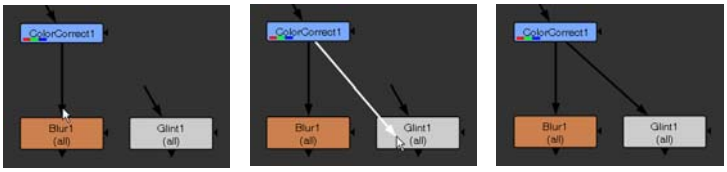
Figure 2-12: Reconnecting a pipe.

Note

Nuke distinguishes the dual inputs that may run into a merge node with the labels **A** and **B**. **A** refers to the foreground element, and **B** to the background element. Nuke always copies from the **A** input to the **B**. This means that if you later decide to disable the node associated with an **A** input, the data stream will keep flowing, because it will, by default, use the **B** input.

To duplicate a connecting arrow:

Shift+drag the connecting arrow on top of the node you want to create a connection to. Nuke duplicates the connecting arrow, leaving the original connection untouched.



To add a node between two connected nodes:

Drag the node into the space between the already connected nodes. As you do so you will see the link between these two nodes become active. When that happens, simply release the node you are dragging and it will automatically wire itself into the network between the two nodes.

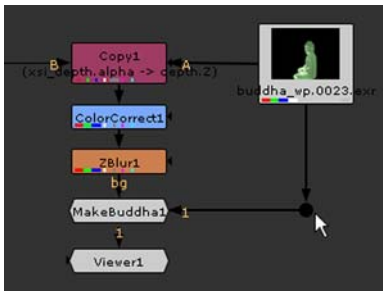
To bend connecting arrows:

1. Select the node before the connector you want to bend.
2. From the toolbar, select **Other** > **Dot**. A dot appears after the selected node, causing a bend in the connector.

3. Drag the dot as necessary to reposition the bend.

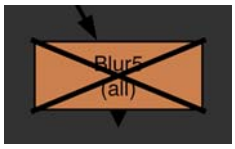
Tip

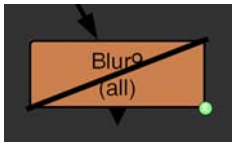
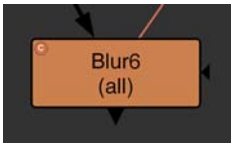
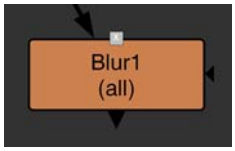
You can also add a dot to an existing connection by pressing **Ctrl** (**Cmd** on a Mac) and clicking on the yellow dot that appears on the connecting arrow.



Indicators on Nodes

There are several indicators that can appear on the nodes in the Node Graph, depending on what you are doing. The following table describes what each indicator means.

Indicator	Where it appears	What it means
		The wide rectangles indicate the channels the node processes. The thin rectangles indicate the channels that are passed through the node untouched.
		The node's effect is limited by a mask from the either the node's primary input or output.
		The node has been disabled by pressing **D** or clicking the **Disable** button.

Indicator	Where it appears	What it means
		The node has been disabled using an expression.
		The node has been cloned. The indicator appears on both the parent and the child node.
		One or more of the node parameters are animated over time.
		One or more of the node parameters are being driven by an expression.
		You are working with a multiview project and have split off one or more views in the node's controls.
		The full effect of the node is not in use, because you have adjusted the **mix** slider in the node's controls.

Searching for Nodes

Nuke's Search feature allows you to search for nodes in your script and select any matches found. As a search string, you can enter all or part of a node name. For example, you can search for all Blur nodes in your script by entering **bl** as the search string.

Using regular expressions, you can also do more complex searches, such as searching for all the Read and Write nodes in a script.

To search for nodes:

1. Select **Edit** > **Search** (or press /) to bring up the search dialog.
2. In the search field, enter the string you want to search for.

   If you want to search for all nodes in the script, enter * (an asterisk).

   If you want to search for all Read nodes in the script, enter **Read**.

   If you want to search for all the Read and Write nodes, enter the following expression:

   (*Read*|*Write*)
3. Click **OK**.

Nuke searches for the nodes in the script and selects any matches it finds.

Note

When you enter expressions in the search dialog, remember that the search field only takes regular expressions. Any characters that have specific meanings in regular expressions, such as [and], need to be preceded by the \ character.

Viewing Information on Nodes

You can obtain more detailed information from any node by selecting that node and then pressing the 'i' key. This will display an information window associated with that node, particularly useful when troubleshooting.

Customising the Node Display

You can modify the colour, name, and notes that a particular node displays. Doing so can make it easier for other artists to decipher the intent of your script. For example, you might colour all nodes green that relate to keying.

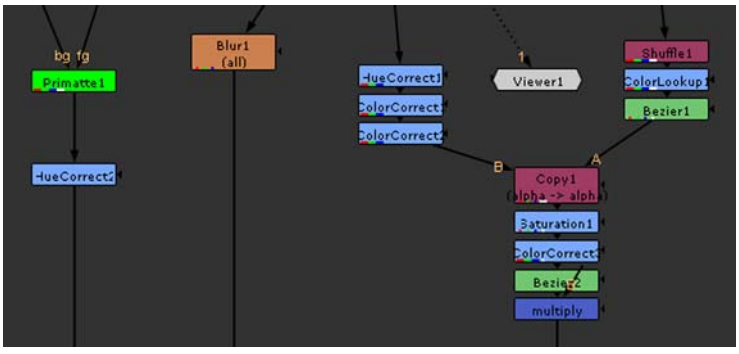


Figure 2-13: Nodes can be colour-coded according to function.

To modify a node's display characteristics:

1. Double-click on the node to display its properties panel.

2. Click the **Node** tab at the top of the dialog. Its attributes appear.
3. Do any of the following:
   - Enter a new name for the node on top of the old name.
   - Click on the left colour button to change the colour of the node. To copy a colour of one node to another, drag and drop the colour button from the node whose colour you want to copy on top of the colour button in the node whose colour you want to change.

   - Type any comments regarding the node in the **label** field. These will appear on the surface of the node.
   - From the **font** pulldown menu, select the font type for any text on the node.
   - Use the buttons on the right to bold or emphasize the text.
   - Enter the font size in the font size field.
   - Click **color** to choose a new font colour.
   - The *Select color* dialog appears, allowing you to select the desired colour.

Tip

You can also have Nuke automatically colour-code nodes for you based on their function. Select **Edit** > **Preferences** (or press **Shift+S**) to open the Preferences dialog. Click on the **Node Colors** tab, and check the **autocolor** option.

Thereafter, every time you add a node to a script, the node will take on the colour appropriate to its function. You can edit these colours as necessary on the **Node Colors** tab of the Preferences dialog.

   - Check **hide input** to conceal the node's incoming pipe. This can enhance the readability of large scripts.
   - Check **postage stamp** to display a thumbnail render of the node's output on its surface.

Navigating Inside the Node Graph

Panning

As scripts grow in complexity, you need to be able to pan to a particular cluster of nodes quickly. The Node Graph offers a couple of methods for doing so.

Panning with the Mouse

To pan with the mouse, press **Alt** and drag the mouse pointer over the workspace. The script moves with your pointer.

Note

On Linux, **Alt**+drag may not work as expected. This is due to the default window functionality on Gnome. To get around it, you can use the **Windows** key instead of **Alt** when panning.

Alternatively, you can change your window preferences on Gnome to fix the problem:

1. Select **Applications** > **Preferences** > **Windows** to open the *Window Preferences* dialog.
2. Under **Movement Key**, select **Super ("or Windows logo")**.

   You should now be able to pan with **Alt**+drag.

Panning with the Map

If your script spills over the edges of the workspace, a navigator map automatically appears in the bottom-right corner.

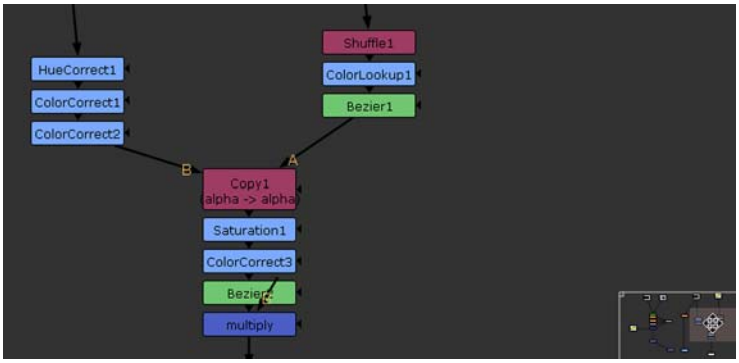


Figure 2-14: Panning with the map.

The map shows you a miniature view of the entire script and the pink rectangle shows the portion of the script that you see within the workspace borders.

To pan with the map, drag the pink rectangle to pan to a different view of the script.

When you pan or resize the window, the map automatically appears when the workspace does not show the entire script. When the whole script is contained within the window border, then the map automatically disappears.

Tip

The navigation map is resizeable. Drag on its upper left corner to make it as large or small as you like.

Zooming

You can zoom in on or out from the script in a couple of ways.

To zoom in:

Move your mouse pointer over the area you want to zoom in on, and press the plus key (+) repeatedly until the workspace displays the script at the desired scale.
OR
Press **Alt** and drag right while holding down the middle mouse button.

To zoom out:

Move your mouse pointer over the area you want to zoom out from, and press the minus key (-) repeatedly until the workspace displays the script at the desired scale.
OR
Press **Alt** and drag left while holding down the middle mouse button.

Note

On Linux, **Alt**+middle drag may zoom the entire Nuke window instead of the Node Graph. This is default functionality on Gnome. To get around it, you can use the **Windows** key instead of **Alt** when zooming.

Alternatively, you can change your window preferences on Gnome to fix the problem:

1. Select **Applications** > **Preferences** > **Windows** to open the *Window Preferences* dialog.
2. Under **Movement Key**, select **Super ("or Windows logo")**.

   You should now be able to zoom in and out of the Node Graph with **Alt**+middle drag.

Fitting Selected Nodes in the Node Graph

To fit selected nodes in the Node Graph, press **F**.

Fitting the Node Tree in the Node Graph

To fit the entire the node tree in the Node Graph, click on the Node Graph to make sure no nodes are selected and press **F**.

Properties panels

When you insert a node, its properties panel appears in the Properties Bin with options to define the node's output. You can also open the properties panel later by doing any of the following:

- Double-click on the node in the Node Graph.
- Ctrl+click on the node in the Node Graph (Mac users Cmd+click).
- Select the node in the Node Graph and press Return.

Tip

To open a properties panel in a floating window, Ctrl+Alt+click (Mac users Cmd+Alt+click) on the node.

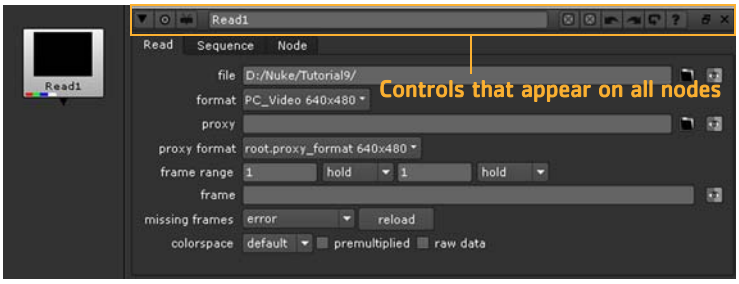
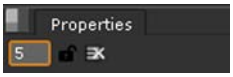


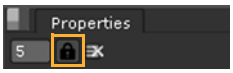
Figure 2-15: The Read node and its properties panel.

Managing the Properties Bin

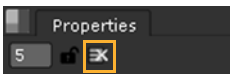
You can limit the number of properties panels that can be open in the Properties Bin. To do so, enter the maximum number of properties panels in the field on the Properties Bin.



To lock the Properties Bin and have all new properties panels appear in floating windows, click the lock button on the Properties Bin.



To empty the Properties Bin and close all the properties panels in it, click the remove all panels button.



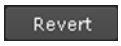
Controls That Appear on All Properties Panels

These are the standard controls of every properties panel:

Control	Function
	Hide or show the node's tabbed pages.
	Centres the node in the Node Graph.
	Centres one of the node's inputs in the Node Graph. Select the input from the pull-down menu that appears.

Control	Function
name field (for example, **Blur1**)	You can enter a new name for the node here.
(left)	Changes the colour of the node. You can drag and drop this button on top of another colour button to copy the colour. To revert to the default colour defined in your Preferences, right-click on the button and select **Set colour to default**. An X on the button indicates the colour is unset, and the colour defined in the Preferences is used.
(right)	Changes the colour used for the node's controls in the Viewer. You can drag and drop this button on top of another colour button to copy the colour. To revert to the default colour defined in your Preferences, right-click on the button and select **Set colour to default**. An X on the button indicates the colour is unset, and the colour defined in the Preferences is used.
	Undoes the last change made to the node.
	Redoes the last change undone.
	Reverts any changes made after the properties panel was opened.
	Displays a pop-up help related to the node and its controls.
	Floats the properties panel. Clicking this button again docks the properties panel back in the Properties Bin if the Bin exists.
	Closes the properties panel.

Floating control panels also include the following buttons:

Control	Function
	Reverts any changes made after the properties panel was opened.
	Reverts any changes made after the properties panel was opened and closes the properties panel. Hitting this button right after a node was created also deletes the node from the Node Graph.
	Closes the properties panel.

Many properties panels also contain several tabbed pages.

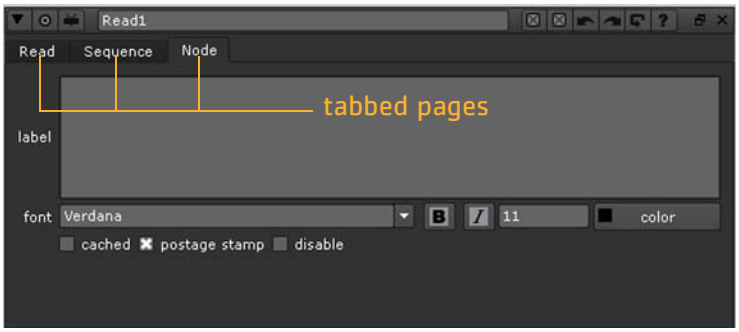


Figure 2-16: The Read node's properties panel and the Node tab.

On the **Node** tab, you can usually adjust the following controls:

Control	Function
label	Lets you add comments to the node. The comments are displayed on the node's surface.
font	Lets you change the font for any text displayed on the node.
	Bolds any text displayed on the node.
	Emphasizes any text displayed on the node.
	Lets you change the font size of any text displayed on the node.
	Lets you change the colour of any text displayed on the node.
hide input	Check this to hide the node's incoming pipe. This control does not appear on all nodes.
cached	Check this to keep the data upstream from the node in memory, so that it can be read quickly.
postage stamp	Check this to display a thumbnail render of the node's output on its surface.
disable	Check this to disable the node. Uncheck to re-enable. (You can also disable or re-enable a node by selecting it in the Node Graph and pressing D.)

Displaying Parameters

To display a node's parameters:

Double-click the node. It's properties panel appears.

Figure 2-27 shows the controls available for editing parameters. Note that the presence of each control will vary according to the parameter's function.

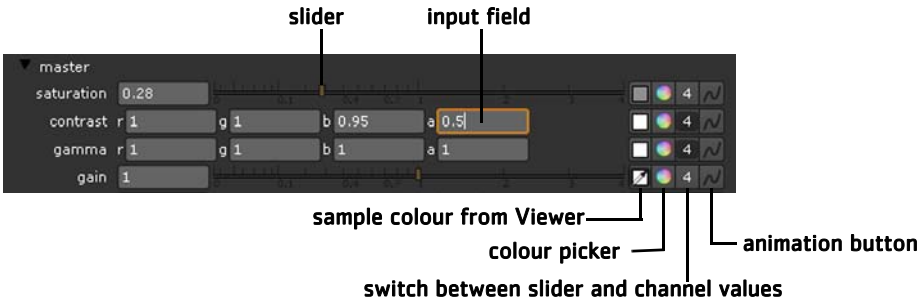


Figure 2-17: Parameter controls.

Using Input Fields

You can key values directly into a field, or use the arrow keys to increment and decrement values.

To key in field values:

1. Click in the field to select the value you wish to replace:
   - Double-click to the left of the value to select only the whole number digits.
   - Double-click the right of the value to select only decimal digits.
   - Press **Return** or double-click on the decimal itself to select all digits.
2. Type the value you want to replace the selection.

Tip

You can also enter expressions (programmatic instructions for generating values) into fields. You always start an expression by typing **=**. See Chapter 16, *Expressions,* on page 357 for information about how to format expressions.

Tip

Nuke also allows you to enter formulas into fields, making it easy to do quick calculations. For example, if you wanted to halve a value of 378, you could simply type **378/2** into a field and press **Enter** to get 189.

You can increment or decrement field values by hundreds, tens, tenths, hundreths, and so on. The magnitude of change depends on the initial position of your cursor. For example if you wanted to increment the initial value of **20.51** by ones, you would insert your cursor before the **0**.

To increment and decrement a field value:

1. Click to insert the cursor just prior to the digit you want to increment or decrement.

2. Press the **up arrow** to increment by unit, or the **down arrow** to decrement by one unit.

Tip

You can also increment and decrement values using the mouse wheel (if available) or by pressing **Alt** while dragging on the value. The latter method is particularly useful for tablet users.

Using Sliders

To set a value with a slider:

Drag the knob to the desired value.
OR
Click the desired value on the graduated scale beneath the slider.

To reset a slider to its default value:

Ctrl+click (Mac users **Cmd**+click) on the slider.

Separating Channels

By default, many parameters in Nuke automatically gang channels for you. For example, if you drag on the Gain slider in the ColorCorrect node, you simultaneously affect the R, G, and B channels (assuming you're processing the RGB channel set). You can, however, use the a parameter's channel chooser button to reveal and edit the values assigned to individual channels.

To edit an individual channel's value:

1. Click the parameter's channel chooser button. The number on its surface lets you know how many channels are available for editing. A series of input fields—one for each individual channel—appears.

2. Edit the values in any of the revealed input fields as necessary.

You can also use a colour slider for editing individual channels. See *Using Colour Sliders and Colour Wheel* on page 52.

Using the Colour Picker and Colour Controls

Nuke offers a colour picker for inputting values. Colour pickers are especially useful for setting white or black points, or for colour matching tasks, such as sampling values in a background plate to colour-correct a foreground image.

To use the colour picker and colour controls:

- Click the colour picker button to display and make colour selections. The colour picker window opens.

- To activate the eye dropper and sample colours from the Viewer (see below), click the colour swatch button.

  To copy a colour from one colour swatch to another, drag and drop the colour swatch with the desired colour on top of the swatch whose colour you want to change.
- To toggle between using the slider and manually entering values for each of the channels, click the channel chooser button (which displays the number of available channels).

Figure 2-18: Options for colour selection.

To sample a colour from the Viewer:

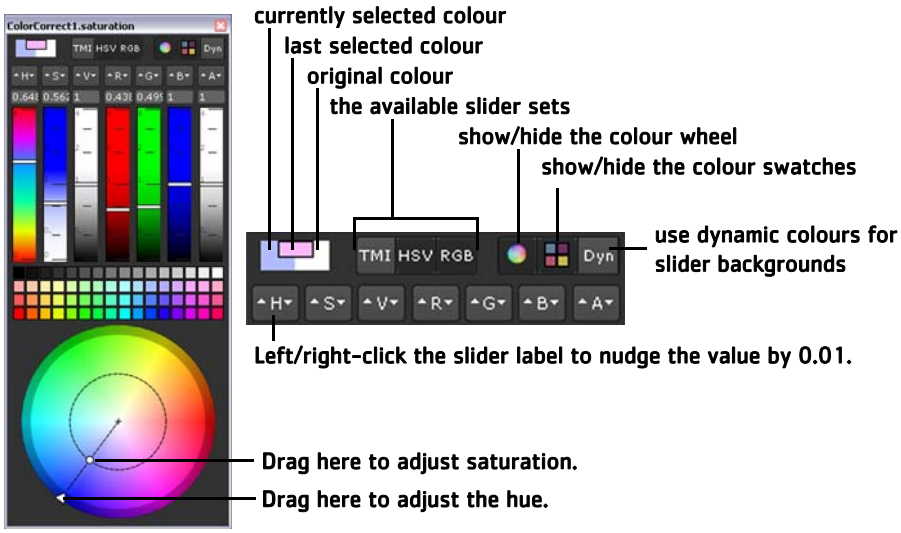
1. In the parameters, click the colour swatch to activate the eye dropper.
2. Move your mouse pointer over the Viewer from which you wish to copy colour values.
3. Zoom and pan as necessary until the region from which you want to sample is clearly visible.
4. **Ctrl/Cmd**+click to sample a colour value from the Viewer, or **Ctrl/Cmd**+**Alt**+click to sample a colour from the node's input while viewing its output. To sample a region rather than a single pixel's colour value, also press **Shift**. The input fields associated with the colour picker update to reflect the colour values of the sampled pixels. In the case of a sampled region, Nuke inserts the average colour values of all inlying pixels.
5. If you didn't manage to sample just the right pixel or quite the right region, **Ctrl/Cmd**+click, **Ctrl**+**Alt**+click, **Ctrl/Cmd**+**Shift**+click, or **Ctrl/Cmd**+**Alt**+**Shift**+click again. A new overlay appears and the old one disappears.
6. When you've captured the right values, click the colour picker button again. The colour swatch reappears displaying the colour or colour average you sampled.

Using Colour Sliders and Colour Wheel

Some parameters offer colour sliders and a colour wheel for inputting values. These offer an intuitive and precise means for assigning just the right colour value to a parameter.

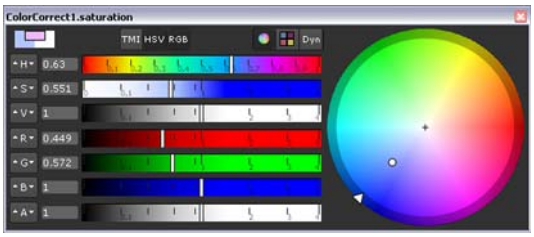
To display the colour sliders and colour wheel:

Click the parameter's colour picker button. The colour sliders and wheel appear.

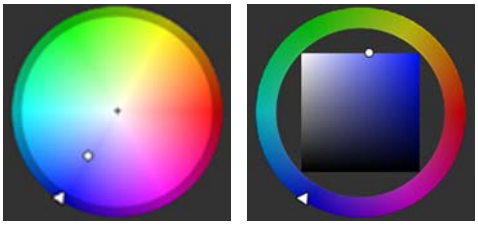

currently selected colour
last selected colour
original colour
the available slider sets
show/hide the colour wheel
show/hide the colour swatches
use dynamic colours for slider backgrounds
TMI HSV RGB
Dyn
Left/right-click the slider label to nudge the value by 0.01.
Drag here to adjust saturation.
Drag here to adjust the hue.

To customise the colour sliders and wheel window:

- From the **TMI**, **HSV**, and **RGB** buttons, select which slider set you want to display. The available slider sets are described below.
- To make the sliders horizontal rather than vertical, resize the colour sliders window. When the window is wide enough, the sliders become horizontal.



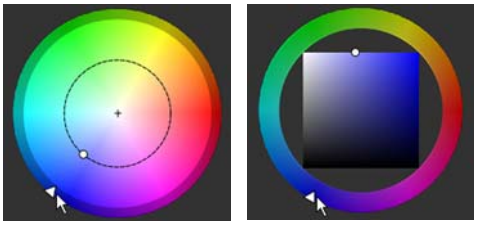
- If you want the background of the sliders show what the value of the colour would be if the sliders were set to the current position, click the **Dyn** button.
- Click the colour wheel button to cycle through three states: the colour wheel, the colour square, and hide colour wheel/square.



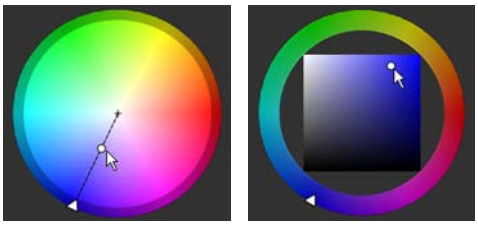
- To hide or show the colour swatches, toggle the colour swatches button.

To use the colour wheel:

- To adjust the hue, drag the marker on the edge of the colour wheel/on the circle.



- To adjust the saturation, drag the marker inside the wheel/square.



Tip

You can also **Ctrl**/**Cmd**+click on the colour wheel to only affect the hue of the selected colour, and **Shift**+click to only affect the saturation.

- To zoom in and out of the colour wheel, press **Alt** and drag right or left with the middle mouse button.
- To pan in the colour wheel, press **Alt** and drag the mouse pointer over the colour wheel.
- To reset the zoom and/or pan, middle-click on the colour wheel.

To use the colour sliders:

To increment the value by 0.01, right-click on the slider label (e.g. **R** or **A**). To decrement the value by 0.01, left-click on the label. Use **Shift**+click for 0.1, and **Alt**+click for 0.001.

You can also click and drag right or left on a label to scrub the value up or down. Use **Shift**+drag to scrub faster, and **Alt**+drag to scrub slower.

These are the functions of the TMI sliders:

- **The temperature slider (T)** lets you control apparent colour temperature by inversely affecting red and green values (assuming your are processing the RGBA channel set). To cool (that is, increase the blue channel's value, while decreasing the red channel's), drag up. To heat (increase the red channel's value, while decreasing the blue channel's), drag down.
- **The magenta/green slider (M)** lets you control the mix of green and magenta hues. To add more magenta (increase the red and blue channel's values, while decreasing the green channel's), drag up. To add more green (increase the green channel's value while decreasing the red and blue channels'), drag down.

- **The intensity slider (I)** lets you simultaneously control the red, green, and blue channel values.
  To increase the value of all channels by the same amount, drag up. To decrease the value of all channels by the same amount, drag down.
  To increase the channel's value, drag up. To decrease it, drag down.

These are the functions of the HSV sliders:

- **The hue slider (H)** lets you control the colour's location on the traditional colour wheel (for example, whether the colour is red, yellow, or violet).
- **The saturation slider (S)** lets you control the intensity or purity of the colour.
- **The value slider (V)** lets you control the brightness of the colour (the maximum of red, green, and blue values).

These are the functions of the RGB sliders:

- **The red slider (R)** lets you control the red channel's value (or the first channel in a channel set if you are processing another set besides RBGA).
  To increase the channel's value, drag up. To decrease it, drag down.
- **The green slider (G)** lets you control the green channel's value (or the second channel in a channel set if you are processing another set besides RBGA).
  To increase the channel's value, drag up. To decrease it, drag down.
- **The blue slider (B)** lets you control the blue channel's value (or the third channel in a channel set if you are processing another set besides RBGA).
  To increase the channel's value, drag up. To decrease it, drag down.

The alpha slider is included in all three slider sets:

- **The alpha slider (A)** lets you control the alpha channel's value (or the fourth channel in a channel set if you are processing another set besides RBGA).
  To increase the channel's value, drag up. To decrease it, drag down.

To use the colour swatches:

When you have found a good colour, you may want to save it in one of the colour swatches for further use. To do so, adjust the colour wheel or sliders until you are happy with the colour, and right-click on the colour swatch where you want to save it. You can also drag and drop a colour into a colour swatch from any other colour button or swatch.

To open another colour picker window:

To open another colour picker window while keeping the first window open, **Ctrl**/**Cmd**+click on another parameter's colour picker button.

To switch between the current and previous or original colour:

The rectangle above the sliders shows the original colour (on the right) next to the currently selected colour (on the left). When you drag the markers to adjust the colour, the last selected colour is shown in between these. To switch between the currently selected colour and the original colour, click on the rectangle.

Animating Parameters

Animating a parameter refers to changing its value over time. You do so by setting key frames (frames at which you explicitly define a value) and allowing Nuke to interpolate the values in between. You can animate most of Nuke's parameters in this manner.

Working with Animated Parameters

The Animation menu lets you set key frames, delete keys, and perform other editing operations on the curves for animated parameters.

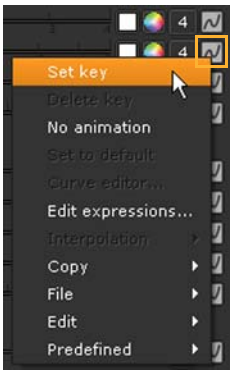


Figure 2-19: The animation menu.

To set key frames:

1. Use a Viewer to cue to a frame where you want to place a key.
2. Click the animation button next to the parameter you want to animate.

3. Select **Set key** from the drop down menu. The parameter's input field turns turquoise, indicating that a key frame has been inserted. Nuke enters the autokey mode: when you change the parameters value at another frame, it will automatically create a key frame for you.

   You can also set a key for all the controls in a node. To do so, select **Set key on all knobs** from the properties panel right-click menu.
4. Cue to the next frame where you want to place a key.
5. Edit the parameter's value using the input field, regular slider, or colour slider. The moment you change the value, Nuke creates a key frame.
6. Continue adding key frames as necessary.
7. Use the Viewer's scrubber to preview the result.

To delete a single key frame:

1. Use the Viewer's next key frame and previous key frame buttons to cue to the key frame that you want to remove. Notice that the scrub bar indicates key frames with a green mark.

2. Click the animation button.

3. Select **Delete key** from the drop down menu.

   Nuke removes the key frame.

To delete all key frames from a parameter:

1. Click the animation button.
2. Select **No animation** from the drop down menu. A confirmation dialog appears. Select **Yes**.

   Nuke removes all key frames from the parameter, and sets the static value to match that of the current frame.

Animated Parameters and the Curve Editor

As you add key frames to a parameter, Nuke automatically plots a curve on its Curve Editor panel, where each value (the y axis) is plotted as it changes over time (the x axis). You can add key frames, delete key frames, and even adjust the interpolation between key frames without ever looking at this curve. However, as the animation grows more complex, you may find it easier to edit the animation by manipulating this curve directly. For more information on how to do so, see *Using the Curve Editor* below.

Using the Curve Editor

Displaying Curves

To reveal an animation curve:

1. Click the animation button next to the parameter whose curve you wish to view.

2. Select **Curve editor**. The Curve Editor panel appears with a focus on the selected parameter's curve.

   The vertical, or y axis, denotes the value of the parameter.

The horizontal, or x axis, denotes time (in frame units).

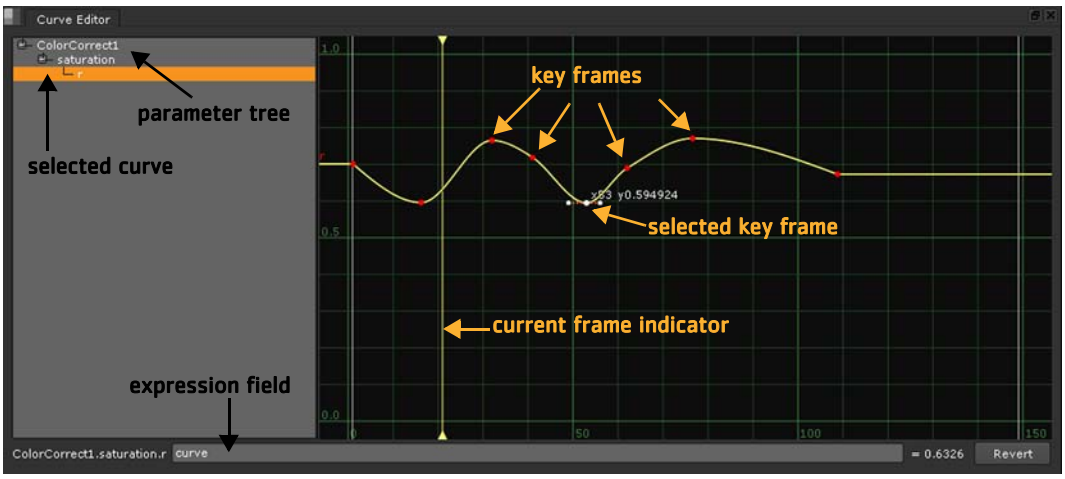


Figure 2-20: Nuke's Curve Editor panel.

To display curves in the Editor:

1. In the parameter tree on the left, click the + and - signs to expand and collapse the hierarchy as necessary.
2. Click a parameter's name to make its curve the focus of the editor. To focus on multiple curves at the same time, **Shift**+click on the names in the parameter tree.
3. To display separate curves for each channel, separate the channels for the relevant control in the node's properties panel.

The parameter tree on the left lets you focus on any curve in the script.

To remove a curve from the Editor:

1. In the parameter tree on the left, click the + and - signs to expand and collapse the hierarchy as necessary.
2. Select a curve in the parameter tree, and press **Delete**.

To zoom in or out in the Editor:

1. Click on the area you want to zoom in on or out of.
2. Press the + button to zoom in, or the - button to zoom out.

   OR

   Scroll up with the mouse wheel to zoom in, or down to zoom out.

Tip

To zoom to a custom area in the Curve Editor, middle-click on the Editor and drag to select an area with a marquee. When you release the mouse button, the Editor will zoom to fit the selected area in the Editor.

To pan in the Editor:

Alt+drag over the editor.

To reset zoom and panning:

1. Right-click on the Curve Editor.
2. From the menu that opens, select **View** > **Frame All** (or press **A** on the Editor).

Nuke centres the curve in the Editor, resetting the zoom.

To centre a portion of the curve in the editor:

1. Select the points you want to centre in the editor.
2. Right-click on the Editor, and select **View** > **Frame Selected** (or press **F** on the Editor).

Nuke centres the selected portion of the curve in the editor. If no points are selected, Nuke centres the selected curve, or all curves.

Editing Curves

You edit curves by moving the points on the curve to new locations. If necessary, you can add more points to the curve. You can also sketch curves freely, use the usual editing functions such as copy and paste, smooth curves with filtering, interpolate curves, loop, reverse or negate curves, and use expressions to modify curves.

To add points to a curve:

1. Click on the curve you want to edit. The curve turns yellow to indicate it's selected.
2. **Ctrl**+**Alt**+click (Mac users **Cmd**+**Alt**+click) on the part of the curve editor you want to add a point to. You can add points both on the curve and outside the curve.

OR

1. Right-click on the Editor and select **Edit** > **Generate**. The *Generate keys* dialog opens.
2. In the **Start at** field, enter the first frame you want to use as a key frame.
3. In the **End at** field, enter the last frame you want to use as a key frame.
4. In the **Increment** field, enter the frame increment you want to use between the first and the last key frame. For example, if you want every tenth frame to be a key frame, enter **10**.
5. In the last field, enter the value you want to use for y. If you do not enter a value here, the key frames are added to the current curve without modifying the curve shape.
6. Click **OK**.

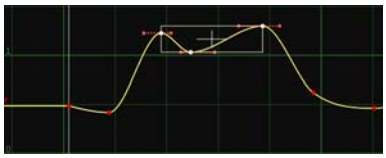
To select points on a curve:

- To select individual points, click on the point you want to select.
- To select multiple points, **Shift**+click on the points, or drag a marquee around them.
  A box is drawn around the points, and the points turn white to indicate they have been selected.
- To select all points, press **Ctrl**+**A** (Mac users press **Cmd**+**A**).
  A box is drawn around the points, and the points turn white to indicate they have been selected.

To move points on a curve:

- To move a point along either the x or y axis only, drag the point to a new location.
- To move a point in any direction, **Ctrl**+drag (Mac users **Cmd**+drag) the point to a new location.
- To adjust the values of a point numerically, select the point and click on the x or y value that appears next to it.

- To move several points at the same time, select them and drag the selection box to a new location.



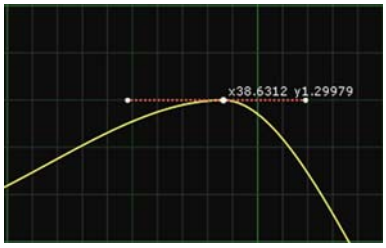
To add or remove points to or from the selection box, **Shift**+click on the points.

To resize and scale the selection box, drag its edges. If the selection box is very narrow, you can press **Ctrl**/**Cmd** when resizing it. This allows you to resize the box in one dimension only. For example, if you have a box that's wide on the x axis but flat on the y axis, you can resize it in this way along the x axis.

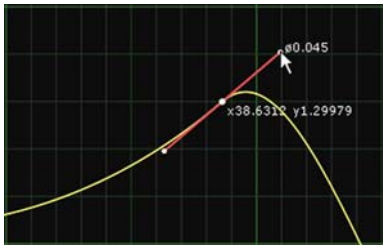
To avoid accidentally moving a point inside the selection box, press **Ctrl**/**Cmd**+**Shift** when dragging the box to hide the points inside the box.

To adjust the slope around the points:

1. Select a point on the curve. Red tangent handles appear on both sides of the point.

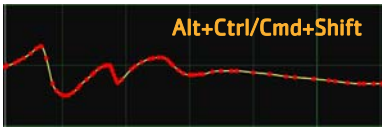


2. Drag the tangent handles to a new location. The curve follows the handles.



To sketch a curve freely:

Press **Alt**+**Ctrl**+**Shift** (Mac users press **Alt**+**Cmd**+**Shift**) while drawing a curve on the editor. Nuke sketches a curve that follows your mouse movements.

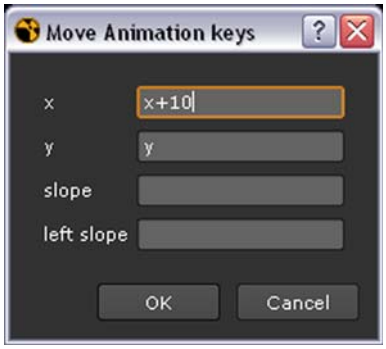


To cut, copy, and paste selected points, expressions, or curves:

1. Right-click on the Curve Editor.
2. From the menu that opens, select **Edit** and the editing function you want to use on the entire curve, for example:
   - **Edit** > **Copy** > **Copy Selected Keys** to only copy the points you have currently selected.
   - **Edit** > **Copy** > **Copy Curves** to copy an entire curve.
   - **Edit** > **Copy** > **Copy Expressions** to copy the expression that creates the curve.
   - **Edit** > **Copy** > **Copy Links** to copy a curve and keep its values linked to the original curve, so that if you change the original, your changes also affect the copied curve.

To move selected points on the curve by a fixed value:

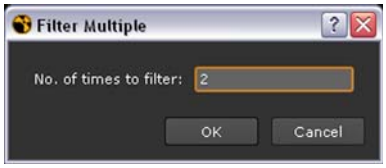
1. Select all the points you want to move.
2. Right-click on the editor and select **Edit** > **Move**. The *Move Animation Keys* dialog opens.
3. In the **x** and **y** fields, define how you want to move the points along the x and y axes. For example, to shift the selected points to the right by a value of 10, enter **x+10** in the x field.



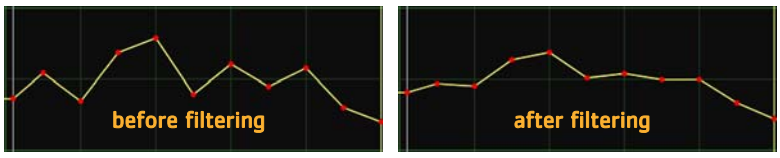
4. In the **slope** and **left slope** fields, define how you want to move the points' tangent handles.

To smooth the curve with filtering:

1. Select the portion of the curve that needs smoothing.
2. Right-click on the editor and select **Edit** > **Filter**. The *Filter Multiple* dialog opens.

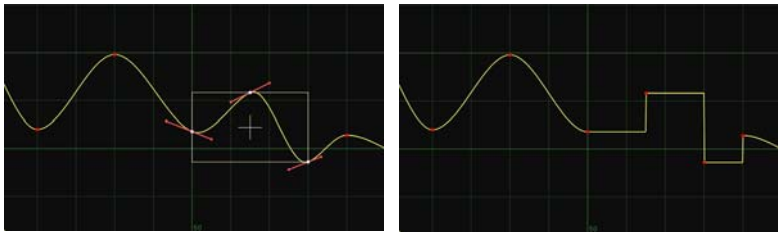


3. In the **No. of times to filter** field, specify how many times you want to filter the curve. Filtering sets new values on each point based on the average values of their neighbouring points. The more filtering, the smoother the curve.

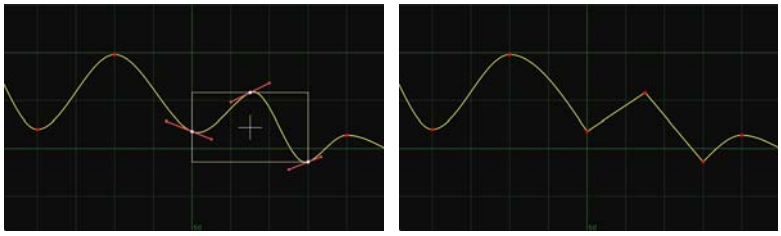


To interpolate parts of a curve:

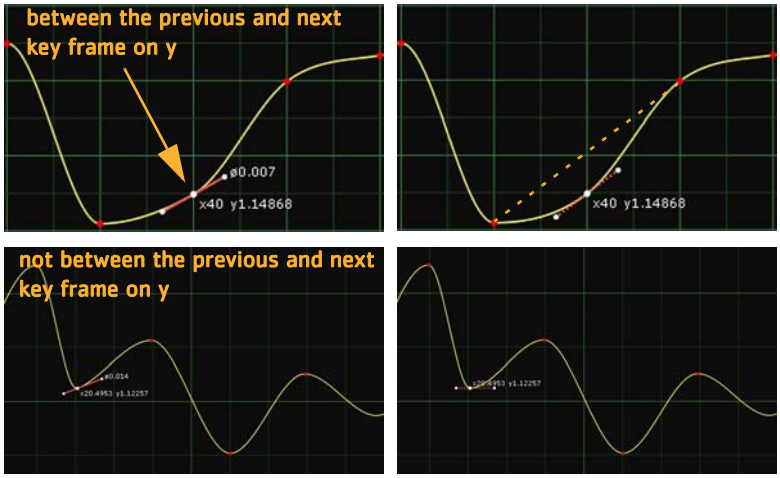
1. Select the point(s) between or around which you want to interpolate the curve.
2. Right-click on the Editor. Select **Interpolation** and the type of interpolation you want to use. Select
   - **Constant** to force a constant value after each selected point.



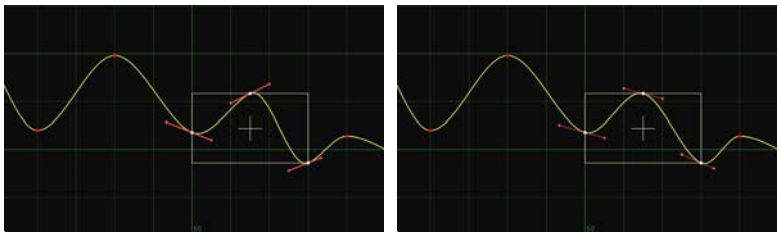
   - **Linear** to use linear interpolation. This produces sharp changes at key frames and straight lines between them.



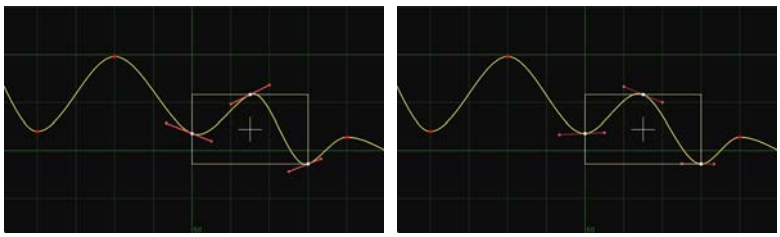
   - **Smooth** to set the tangents' slopes equal to the slope between the key frame to the left and the key frame to the right if the selected point is between these two key frames along the y axis. If the selected point is not between these key frames and has a larger or smaller value than both key frames, the tangents' slopes are made horizontal. This ensures the resulting curve never exceeds the key frame values.



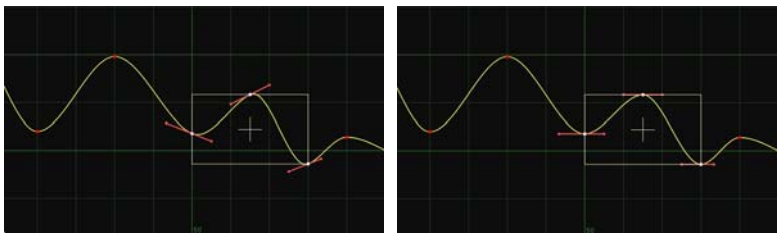
- **Catmull-Rom** to set the tangents' slope equal to the slope between the key frame to the left and the key frame to the right regardless of where the selected point is located. The resulting curve can exceed the key frame values.



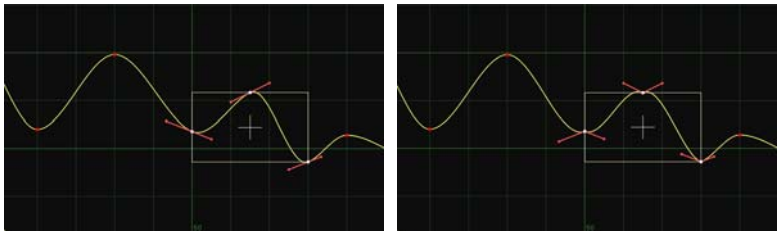
- **Cubic** to set the slope so that the second derivative is continuous. This smoothens the curve.



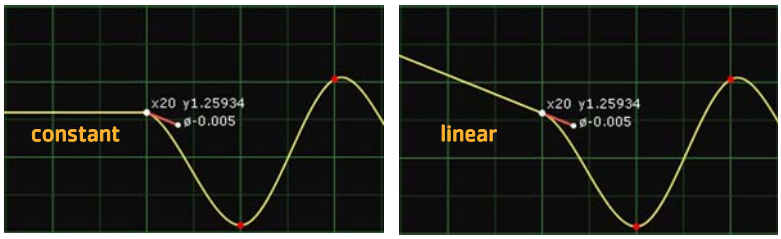
- **Horizontal** to make the tangents horizontal, setting the slope around the selected points to zero.



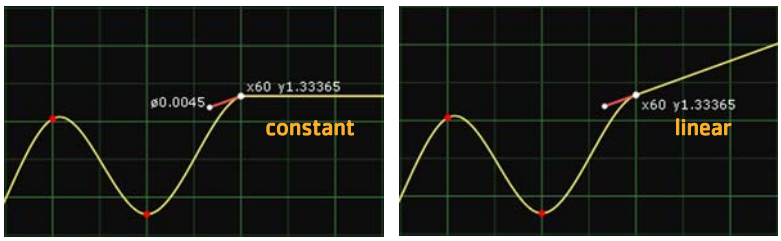
- **Break** to adjust the two tangents of a selected point independent of each other.



- **Before** > **Constant** or **Linear** to interpolate the parts of the curve that are on the left side of the first point. This option only works if you have selected the first point on the curve.

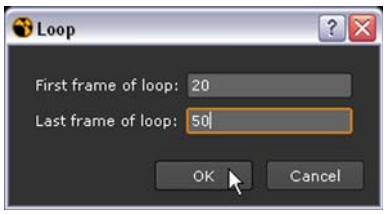


- **After** > **Constant** or **Linear** to only interpolate the parts of the curve that are on the right side of the last point. This option only works if you have selected the last point on the curve.

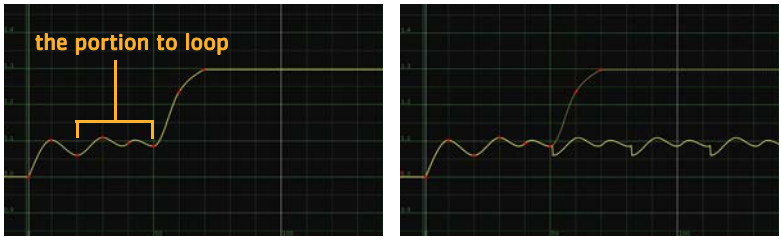


To repeat a portion of the curve throughout the curve:

1. Right-click on the editor and select **Predefined** > **Loop**. The *Loop* dialogue opens.
2. In the **First frame of loop** field, enter first frame of the portion you want to repeat throughout the curve.
3. In the **Last frame of loop** field, enter the last frame of the portion you want to repeat.
4. Click **OK**.



The shape of the curve between these frames is repeated throughout the rest of the curve. The solid line represents the actual curve, and the dotted line the original curve with the key frames.



To reverse a curve:

Right-click on the editor and select **Predefined** > **Reverse**.
This makes the curve go backward in time. Both the new curve and the original curve are displayed. The solid line represents the actual curve, and the dotted line contains the key frames that you can modify.

To negate a curve:

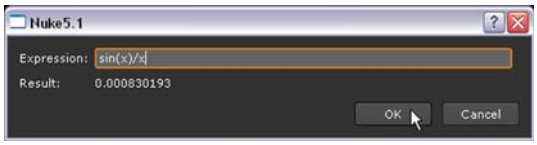
Right-click on the editor and select **Predefined** > **Negate**.
The curve becomes the negative of the key frames. For example, a value of 5 turns into -5. Both the new curve and the original curve are displayed. The solid line represents the actual curve, and the dotted line contains the key frames that you can modify.

To use an expression to modify a curve:

Enter the expression in the expression field at the bottom of the Curve Editor.

OR

1. Right-click on the Editor, and select **Edit** > **Edit expression**.
2. In the dialog that opens, type the expression you want to use for the curve, for example, **sin(x)/x**.



3. Click **OK**.

Viewers

Viewer nodes, unlike process nodes, don't alter data in any way; rather, they act as windows on it. Each Viewer node displays the render output of any connected process nodes in the Viewer panel. Viewer nodes are essential for quickly assigning the right values to parameters because they allow you to edit in context—that is, edit a given node's parameters upstream in a script while viewing the effect of those changes downstream.

You can place as many Viewer nodes in a script as you wish, which allows you to simultaneously view multiple outputs. You can also pipe the output from up to ten process nodes into single Viewer node, and then cycle through the various displays. This allows you to easily compare an image before and after processing by a given effect.

Note

The maximum image size the Nuke Viewer can display is 2^28 = 268,435,456 pixels. This is the same as 16k x 16k, 32k x 8k, 64k x 4k, or 128k x 2k. If your image is larger than this, it will be resized and you will get the following warning:

Viewer image is clipped to <size> x <size>!

For example, if your image resolution is 60,000 x 4473, Nuke is able to display the image because the number of pixels is less than 2^28. However, if the resolution is 60,000 x 4474 (more than 2^28 pixels), the image will be resized to 59998 x 4474.

In addition to the Viewer, this limit is also applied to the bounding box of the images being passed between each node.

Adding Viewer Nodes

Viewers have corresponding nodes that appear in the Node Graph. These nodes do not produce output for rendering; they generate display data only. You can connect Viewer nodes as described in *Working with Nodes* on page 35. In practice, you'll work faster by using the Viewer hotkeys described below.

To add a Viewer node:

1. Select the node whose output you wish to view.
2. Do one of the following:

- Using the menu bar, choose **Viewer** > **Create New Viewer**.
- Using the toolbar, choose **Image** > **Viewer**.
- Using a keyboard shortcut, press **Ctrl+I** (Mac users press **Cmd+I**).

Nuke connects a Viewer node to the node you selected in step 1, and displays the output of the node in the Viewer panel. You can also insert a Viewer node and set up its first connection by simply pressing **1** over the Node Graph.

Connecting Viewer Nodes

Once you add a Viewer node to the script, you can quickly pipe any process node's output to it simply by selecting the process node then pressing any number key. Doing so pipes the output to one of the ten input ports available on every Viewer node (the **0** key represents the tenth slot).

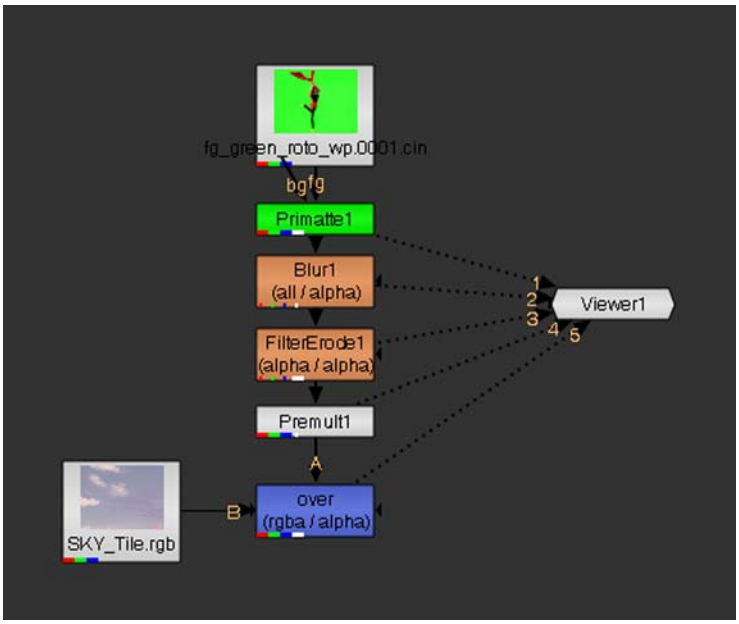


Figure 2-21: A Viewer node with multiple inputs.

Toggling Views

If a Viewer node has multiple inputs, like the one depicted above, you can press the up or down arrow keys to quickly cycle through the views (your cursor needs to be in the Viewer window). To view a particular node press the number key (**1, 2, 3... 0**) corresponding to the pipe number whose contents you wish to view.

Panning and Zooming the Viewer Window

To pan the frame:

Hold the **Alt** key and drag on the display. The frame follows the mouse pointer.

Note

On Linux, **Alt**+drag may not work as expected. This is due to the default window functionality on Gnome. To get around it, you can use the **Windows** key instead of **Alt** when panning.

Alternatively, you can change your window preferences on Gnome to fix the problem:

1. Select **Applications** > **Preferences** > **Windows** to open the *Window Preferences* dialog.
2. Under **Movement Key**, select **Super ("or Windows logo")**.

   You should now be able to pan with **Alt**+drag.

To recentre the frame:

Press **F**.

To zoom in on the frame:

1. Move your pointer over the area of the display on which you want to zoom.
2. Press the plus button (**+**) repeatedly until the frame attains the desired scale.

   OR

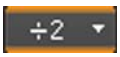
   Select **zoom in** from the zoom pulldown menu in the top right corner.

To zoom out from the frame:

1. Move your pointer over the area of the display from which you want to zoom.
2. Press the minus button (**-**) repeatedly until the frame displays at the desired scale.

   OR

   Select **zoom out** from the zoom pulldown menu in the top right corner.



To restore the zoom to 100%:

Press **Ctrl**+**1** (Mac users press **Cmd**+**1**).

Hiding Floating Viewers

To hide a floating Viewer:

Press ` (the accent key).

To show a hidden floating Viewer:

Press ` (the accent key) again.

Using the Viewer Controls

A Viewer's onscreen controls let you navigate the timeline, display channels, zoom, choose cameras (3D mode), and create display wipes and composites.

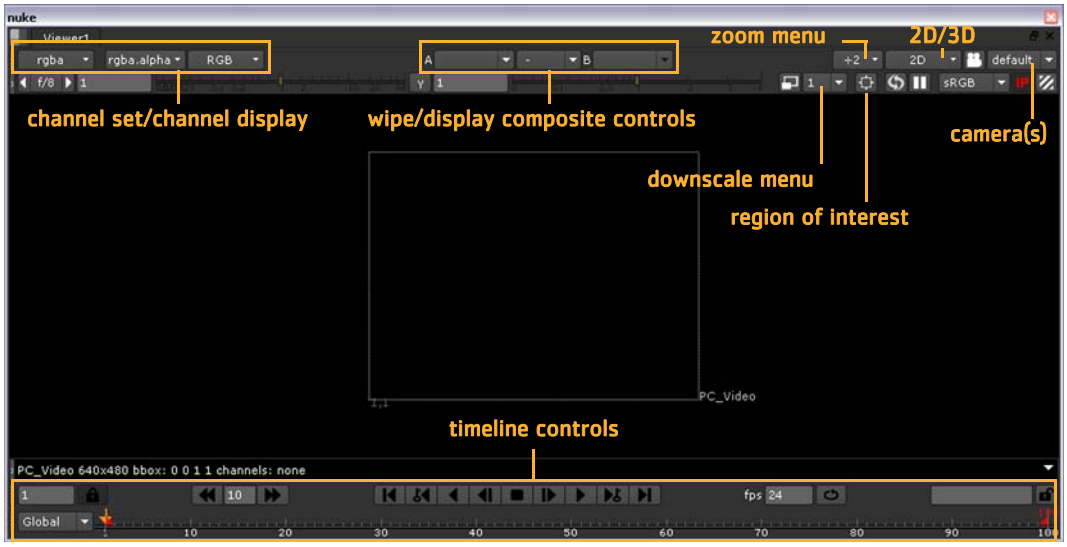


Figure 2-22: Viewer controls.

Timeline Controls

Drag the orange marker along the timeline to quickly cue to a specific frame. The number of the current frame appears in the **Current** field above the timeline. You can also cue to a frame by typing its number directly into this field.

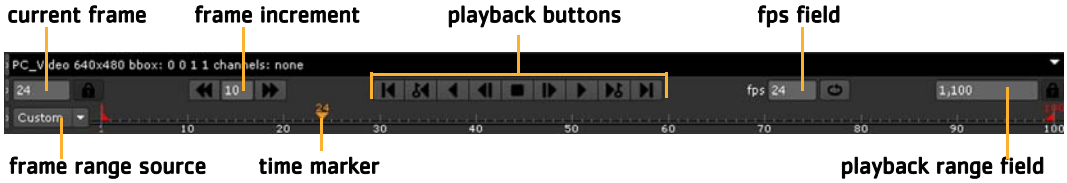


Figure 2-23: Timeline controls.

By default, Nuke automatically adjusts the timeline of every Viewer window to show the frame range defined in your Project Settings. If no frame range is defined, the frame range of the first image you read in is used as the global frame range.

Viewer timeline controls also have a frame range source pulldown menu that you can use to define where the timeline gets its frame range from. You can set this menu to **Global**, **Input**, or **Custom**. **Global** is the default setting described above.

To have the Viewer adjust the timeline to show the "in" and "out" frames of the current input clip, select **Input** from the frame range source menu. The number of the first frame in the clip is shown in the left end of the timeline and the number of the last frame in the right end. If you change the input of the Viewer, the frame range on the timeline is adjusted accordingly.

To manually adjust the frame range for the current Viewer window, pan and zoom on the timeline until you see the desired frame range and **Custom** becomes selected in the frame range source menu. **Alt**+drag to pan, and **MMB**+drag to zoom in. You can also zoom in on or out of the timeline using the mouse wheel. To reset the zoom, press the middle mouse button over the timeline.

To adjust the playback range for the current Viewer window, **Ctrl**+drag (Mac users **Cmd**+drag) the red playback range marker on the timeline to a new "in" and "out" frames as shown in Figure 2-38, or enter a new playback range in the playback range field.

Figure 2-24: Adjusting the frame range for the current Viewer.

The **fps** field (frames-per-second) initially displays the project's playback speed. Nuke will attempt to maintain this speed throughout playback, although this adjusts depending on the resolution of the imagery and your hardware configuration.

The Frame Increment field lets you specify the number of frames by which the Previous increment/Next increment buttons cue the sequence.

The following table lists the functions of the playback buttons:

Buttons	Functions
	The First frame and Last frame buttons cue the sequence to the first and last frame.
	The Previous key frame and Next key frame buttons cue the sequence to the script's previous or next key frame.
	The Play backward and Play forward buttons play the sequence backward or forward at the script's frame rate.
	The Back 1 Frame and Forward 1 Frame buttons cue the sequence to the previous or next frame.
	The Stop button halts playback.
	The Previous increment and Next increment buttons cue forward or back by 10 frames by default. These are useful for heavy keyframing tasks. You can adjust the increment value as necessary.

The Playback Mode button lets you control how many times and in what direction the Viewer plays back the sequence. Click the button to toggle between the following modes:

Button	Function
	Repeatedly play the sequence (loop).
	Play the sequence once from the current frame to the head or tail (stop).
	Repeatedly play the image back and forth from head to tail.

Jumping to a Specific Frame

You can move quickly to a specific frame on the timeline by choosing **File** > **Go to frame** (or by pressing **Alt**+**G**), entering a frame number in the dialog that appears, and clicking **OK**.

Synchronising Viewer Playback

The Lock/Unlock button lets you toggle synchronized playback of Viewer windows. By default, all Viewers are locked—that is, if you cue to a frame in one Viewer, all other Viewers follow suit.

Figure 2-25: Synchronizing Viewers.

When the lock icon changes from a closed lock to an open lock, that Viewer's playback becomes independent of other Viewers, and not cued to the other Viewers.

Pausing the Display

The Pause button stops the Viewer from updating and holds the last frame rendered. To reactivate display rendering for all frames, press the button again.

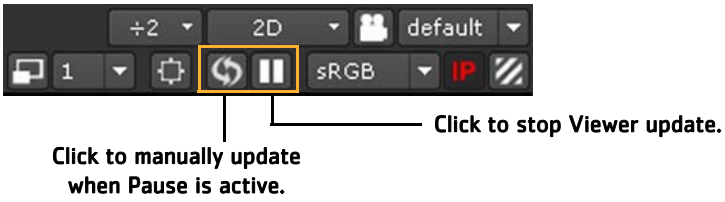


Figure 2-26: Pausing the display update of a Viewer.

You can click the render update button next to Pause (or press **U**) to manually update the display while keeping Pause active.

Displaying a Single Channel

You can press the **R**, **G**, **B**, and **A** keys on your keyboard to display the red, green, blue, and alpha channels respectively. Or, you can also select a channel from the **RGB** pulldown menu in the top left corner.

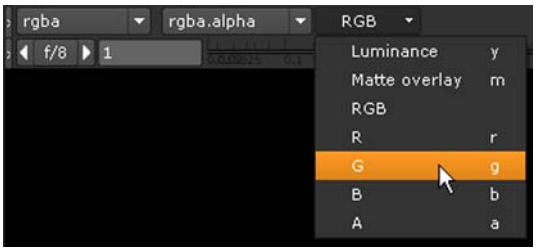


Figure 2-27: Displaying a single channel.

Press one of the channel keys again to toggle back and display all channels.

Tip

If you press **Shift** while selecting the channel, your selection only affects the currently active input of the Viewer node. This way, you can display different channels from the Viewer's different inputs. For example, when keying it can be useful to view the RBG channels from one input and the alpha channel from another, and toggle between the two. To achieve this, do the following:

1. Create a Viewer with several inputs. See *Connecting Viewer Nodes* on page 67.
2. Activate one of the inputs by pressing its number (for example 1) on the Viewer.
3. Press **Shift** and select **RGB** from the channel menu.
4. Activate another input (for example, input 2) by pressing its number on the Viewer.
5. Press **Shift** and select **A** from the channel menu.
6. Toggle between the inputs by pressing their numbers or the up and down arrow keys.

Channel Set and Channel Display Lists

The channel set list lets you choose a set of colour channels to display in the Viewer. By default, this is set to display the rgba set, but you can choose any channel set in the data stream.

Figure 2-28: Selecting a channel set

The Channel list controls which channel appears when you view the "alpha" channel. The default setting actually does display the alpha channel when you press the **A** key; however, you can change this by selecting any channel in the data stream.

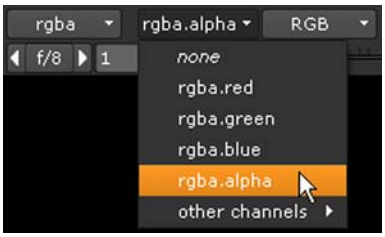


Figure 2-29: Selecting the channel to display when **A** is pressed.

Superimposing an Image's Alpha Channel over its RGB Channels

When you've read in an image that has an alpha channel, you can display the alpha channel as a red overlay on top of the image's red, green, and blue channels.

To display an image's alpha channel on its RGB channels:

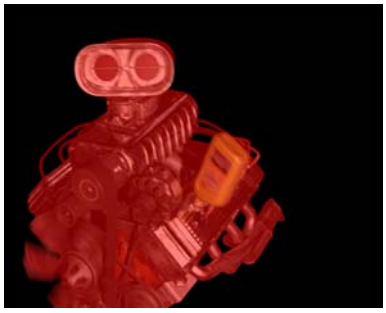
1. Select **Image** > **Read** to read in an image.
2. Connect a Viewer node to the Read node.

   By default, Nuke displays the red, green, and blue channels in the Viewer.

3. Click on the Viewer to make sure it's the currently active panel.

4. Press **M**.

   Nuke displays the image's alpha channel as a red overlay on top of the RGB channels.



5. To return to the RGB display, press **M** again.

Image Format Labels

The Pixel Value indicator displays information about the pixel underlying the pointer or about a sampled pixel or region of pixels. (You can sample a single pixel from the Viewer by pressing **Ctrl/Cmd** while clicking, a region from the Viewer by pressing **Ctrl/Cmd**+**Shift** while dragging, a single pixel from the node's input by pressing **Ctrl**/**Cmd**+**Alt** while clicking, and a region from the node's input by pressing **Ctrl**/**Cmd**+**Alt**+**Shift** while dragging.) From left to right, the indicator displays the following about the current pixel or sample: its x and y position; its Red, Green, Blue, and Alpha values; and other values depending on the colour type you have selected from the colour type menu on the right.

The Format indicator displays the image resolution and the size of the bounding box.

Using the Zoom List

The Zoom list lets you select the magnification factor by which the current image is displayed. This list also shows the hotkeys to press to quickly switch between the different zoom settings.

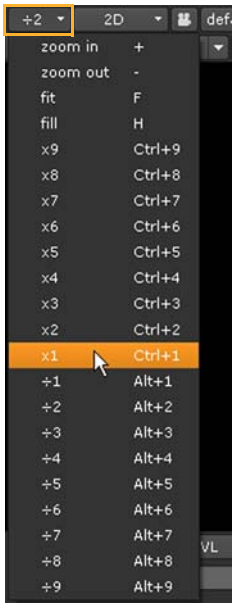
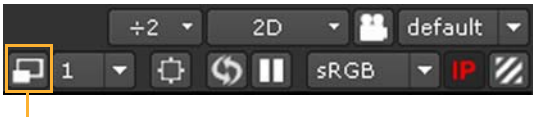


Figure 2-30: Selecting a display zoom.

Proxy Mode

Nuke can generate low-res proxies for displayed frames as needed when you press **Ctrl/Cmd+P** or click the proxy mode toggle button on the Viewer to activate the proxy display mode.



proxy mode toggle

For more information, *Proxy Mode* on page 92.

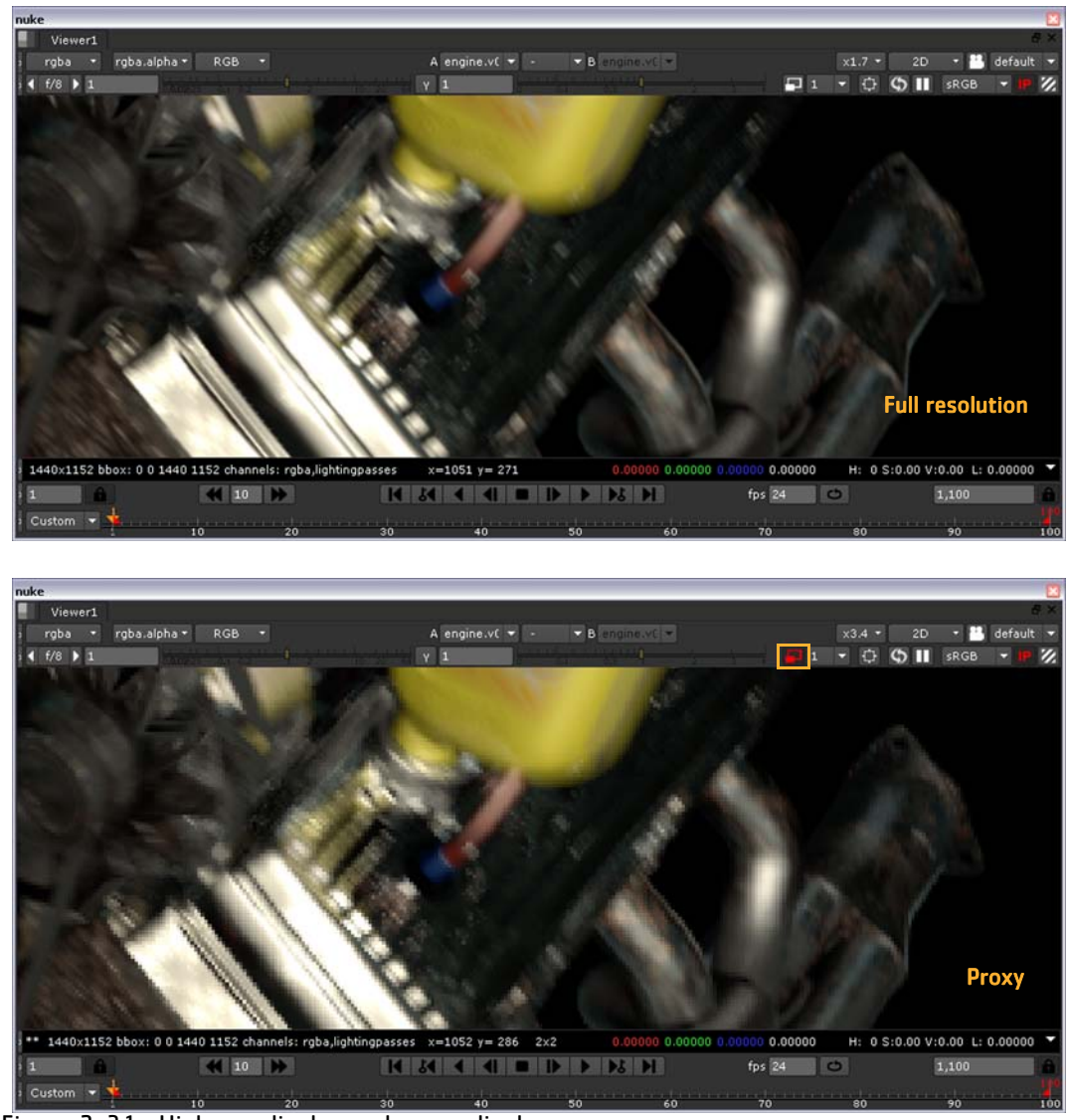


Figure 2-31: High-res display and proxy display.

The global proxy resolution and/or scale are determined by the project settings, which you can open by selecting **Edit** > **Project settings** (or pressing **S**).

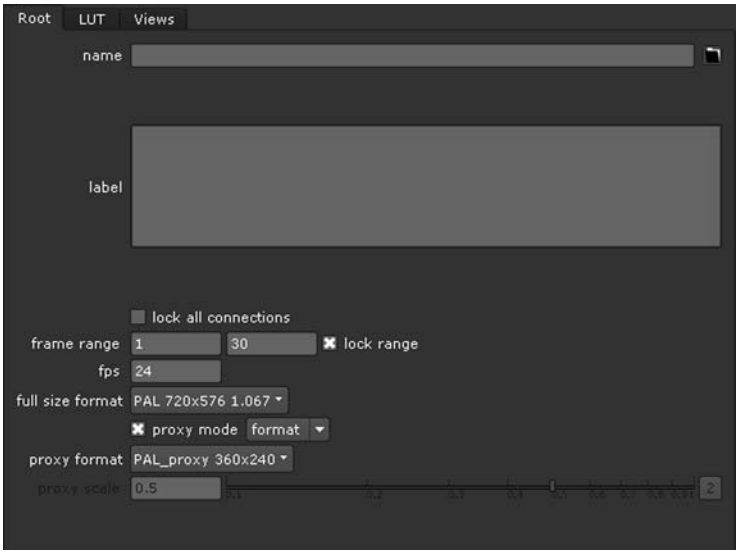


Figure 2-32: Proxy display resolution defined on the project settings properties panel.

You can also read in rendered proxies using the Read nodes' controls. The proxy file does not need to match the proxy resolution in use. The closest larger image (full-size or proxy file) will be scaled down to the required size..

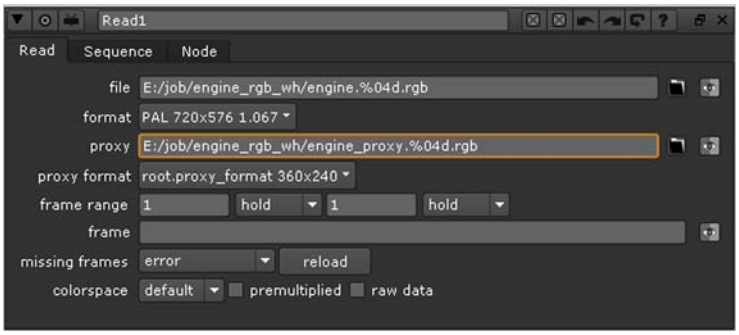


Figure 2-33: Reading in proxy versions of images.

Lowering the Display Resolution of Individual Viewers

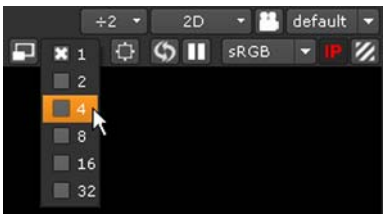
Viewers also have a pulldown menu that allows you to easily switch to lower display resolutions, regardless of whether you have activated proxy mode or not. Using this multiplier setting, you can, for example, change the display resolution of an individual Viewer to 50% of the current (be it full-size or proxy) resolution. This is useful if you want to have Nuke display your images more quickly without having to touch the project settings. It also comes in handy if you have just a few very large plates in your script, as you can choose to use lower resolutions when viewing just these plates.

To lower the display resolution of individual Viewers:

From the Viewer's down-rez menu, choose the factor by which you want to lower the display resolution. Select:

- **1** to display 1/1 of the currently active resolution.

- **2** to display 1/2 of the currently active resolution.
- **4** to display 1/4 of the currently active resolution.
- **8** to display 1/8 of the currently active resolution.
- **16** to display 1/16 of the currently active resolution.
- **32** to display 1/32 of the currently active resolution.



For example, if you have a 4K plate and are using a proxy scale of 0.5, your plate will still be 2K even in the proxy mode. Setting the down-rez factor to **2** in the Viewer will scale the plate down further to 50% of the proxy resolution, that is to 1K. This gives you much faster (but less accurate) feedback.

Pixel Aspect Ratio

The pixel aspect ratio determines whether your images are displayed using square or rectangular pixels. By default, the Viewer uses the pixel aspect ratio defined in your project settings. To see the current setting, select **Edit** > **Project settings** (or press **S**).

For example, a pixel aspect ratio of 2 accurately displays anamorphic footage the way it will be projected, like this:

Figure 2-34: The Viewer uses the pixel aspect ratio defined for the script.

If you want to ignore the pixel aspect ratio, you can toggle it by pressing **Ctrl/Cmd+Shift+P** over the Viewer window.

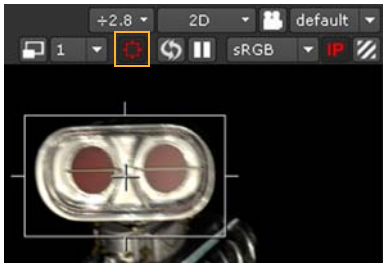
Figure 2-35: Press Ctrl/Cmd+Shift+P over the Viewer window to ignore the pixel aspect ratio.

Region of Interest (ROI)

The **ROI** button lets you enable rendering only through a region of interest—a portion of the image you explicitly select. This is useful for quickly viewing render results in a process-heavy script.

To define a region of interest:

1. Click on the ROI button in the Viewer controls. The ROI overlay appears.



2. Drag to resize and move the ROI overlay as necessary.

OR

1. Over the Viewer, press **Alt+W** once (do not hold the keys down). The ROI button turns red, but the ROI overlay does not appear. This allows you to freely draw your own ROI rather than adjust the default overlay.
2. Drag a marquee to draw the region of interest where you need it.

To clear a region of interest:

1. After you've set a region of interest, you can clear it by pressing **Alt+W** over the Viewer. You can then drag a new marquee to define a new region of interest.
2. To turn off the feature and update the whole Viewer with the recent changes, click the **ROI** button again (or press **Shift+W**).

Adjust Display Gain and Gamma

The gain and gamma sliders let you adjust the displayed image, without affecting your final output. These controls are useful for tasks like spotting holes in mattes. You can boost or reduce gain by entering a multiplier (exposure value), dragging on the slider, or using the F-Stop arrows. Boost or reduce gamma by entering a gamma level or dragging the gamma slider.

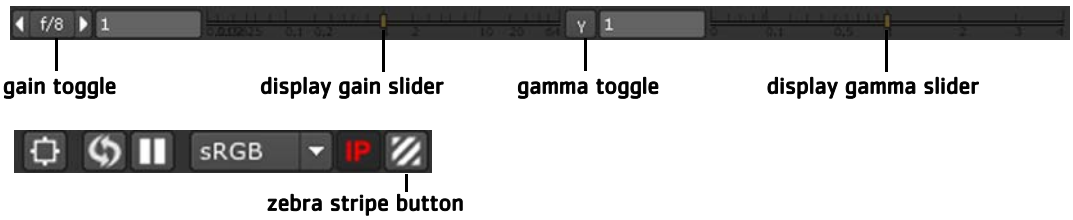


Figure 2-36: Display gain and gamma controls

The gain and gamma toggle buttons let you switch between the default values of 1 (no change) and the last gain and gamma adjustments you made in the Viewer.

The Zebra Stripe button, when pressed, applies stripes to all pixels outside the range 0.0 to 1.0.

Viewer Input Toggle / LUT Toggle

A Nuke script may include a special node—sometimes called a Viewer LUT or lookup table—that adjusts the Viewer display to simulate the way the image will look after transfer to film, video, or other devices or media.

Figure 2-37: Toggling the use of the VIEWER_INPUT node.

Usually, the Viewer LUT is a custom script that you create and import as a node. This must be named VIEWER_INPUT to have an effect on the Viewers. Any image you view is passed through it when the IP button is activated.

Note

Normally, the Viewer will send the image it is going to display through the VIEWER_INPUT process node, then apply gain/gamma and LUT effects to the result prior to display. However, depending on what the input process node is doing, this may not be the correct order. Therefore, if your VIEWER_INPUT process node has float controls named "gain" and/or "gamma", then the Viewer will drive them from the corresponding Viewer controls and not do that image processing itself. This allows you to implement the gain and gamma in your VIEWER_INPUT process gizmo/group node using whatever nodes and order you want. If your input process node does not have gain/gamma controls, then the Viewer will apply the effects in its normal way after running the image through the VIEWER_INPUT process node.

2D / 3D Toggle and Camera Controls

The 2D / 3D list lets you toggle between 2D and 3D display modes in the current Viewer. This list also lets you choose between different orthographic (non-perspective) views when working in the 3D mode.

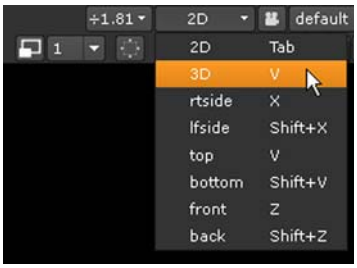


Figure 2-38: The 2D / 3D views and camera controls.

The camera list on the right lets you choose which camera to look through when multiple cameras exist in your 3D scene. For more information on these controls, see Chapter 13, *3D Compositing,* on page 277.

Using the Viewer Composite Display Modes

The wipe control provides an option for displaying a split-screen of two images, which can help you compare before and after versions for colour correction, filtering, and other image manipulation. This control also includes display compositing options to overlay different images.

To display a comparison wipe:

1. Select a node in your script and press **1** to display its output in the Viewer.
2. Select the node you want to compare and press **2**.

   The **2** keystroke connects the image to the Viewer (assigning the next available connection, number 2).
3. From the **A** and **B** lists on top of the Viewer, select the images you want to compare. The lists include the last four nodes connected to the Viewer.
4. From the Viewer composite list in the middle, select **wipe**.

The two images are displayed split-screen in the Viewer.

5. Drag the handles of the crosshair to adjust the wipe:
   - Drag the crosshair centre to change its position.
   - Drag the long handle (on the right) to rotate the wipe.
   - Drag the "arc" handle to cross-dissolve the second image.
6. When finished with the split-screen, select none (–) from the Viewer composite list.

Tip

If you press **Shift** while selecting a channel, your selection only affects the currently active input of the Viewer node. This way, you can display different channels from the Viewer's different inputs. For example, when keying it can be useful to view the RBG channels from one input and the alpha channel from another, and toggle between the two.

The display composite options—**over**, **under**, and **minus**—can also be selected to overlay two images. When the two images are 2D, this allows you to create a quick comp.

When one image is 2D and the other is a 3D node, you can use **under** to line up the wireframe preview with the 2D reference, and see how the 3D matches prior to a full render.

One example of this is when you want to preview a wireframe 3D scene with a background plate that you are trying to match, as shown below.

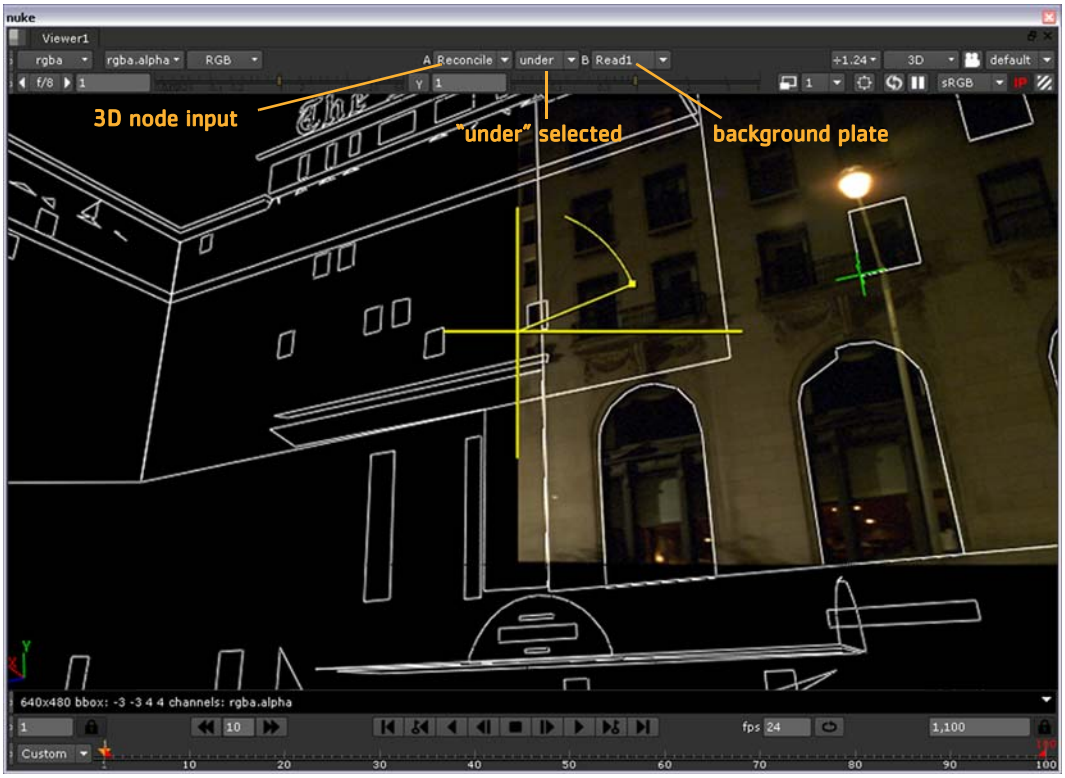


Figure 2-39: Comparing a 3D scene over a 2D image.

For more information, see Chapter 13, *3D Compositing,* on page 277.

Using the File Browser

Whenever you load or save files in Nuke, you'll see a browser similar to the one shown in Figure 2-40. The directory navigation buttons let you create or access the directory from which you wish to read or write data.

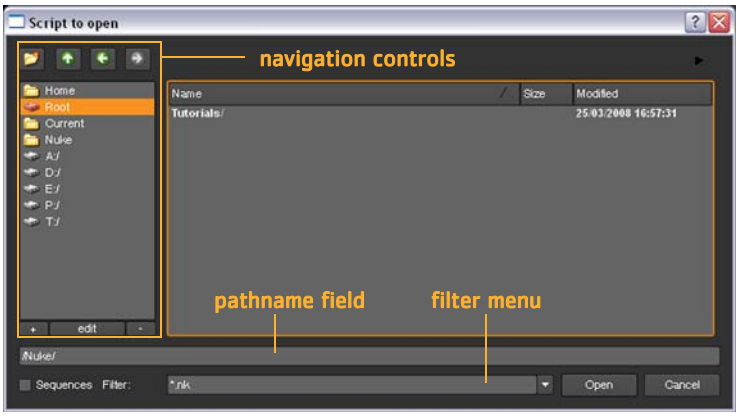


Figure 2-40: Nuke's file browser.

The navigation controls let you move through the directory structure, bookmark favourite directories, and create new directory folders.

To use the navigation controls:

- Click the **Create New Directory** button to create a new directory at your current position in the file hierarchy.
- Click **Up one directory** to ascend one directory level closer to the root.
- Click **Home** to access the directory defined as your local working directory.
- Click **Root** to ascend to the very top of your local drive or server's file hierarchy.
- Click **Work** to access the directory you (or your system administrator) defined as your network working directory.
- Click the **+** button to add a directory bookmark.
- Click the **edit** button to edit the name or pathname to a bookmark.
- Click the **–** button to remove a directory bookmark.

To limit the file list to specific file types, use the **filter** pulldown menu and **Sequences** checkbox.

To use the filter pulldown menu and Sequences checkbox:

- Select ***.nk** to display only Nuke script files.
- Select ***** to display all files (except hidden files), regardless of whether they're associated with Nuke.
- Select **.*** to display all files, including hidden files.
- Select ***/** to display directory names, but not their contents.
- Check **Sequences** to display image sequences as single titles, as in fgelement.%04d.cin 1-50 rather than fgelement.0001.cin, fgelement.0002.cin, fgelement. 0003.cin, and so on.

Pathname Field

The pathname field displays the current directory path, lets you navigate to a new path, and also enter a filename for scripts and rendered images.

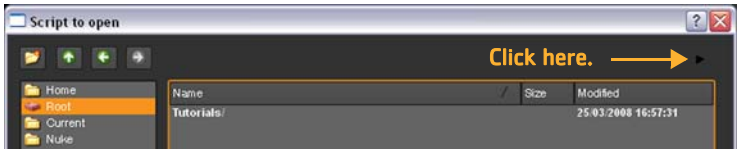
Figure 2-41: Pathname field.

To use the pathname field:

1. To navigate to a directory, type the pathname in the field and press **Tab**.
2. To enter a script name, browse to a directory path and enter the file name after the displayed path.

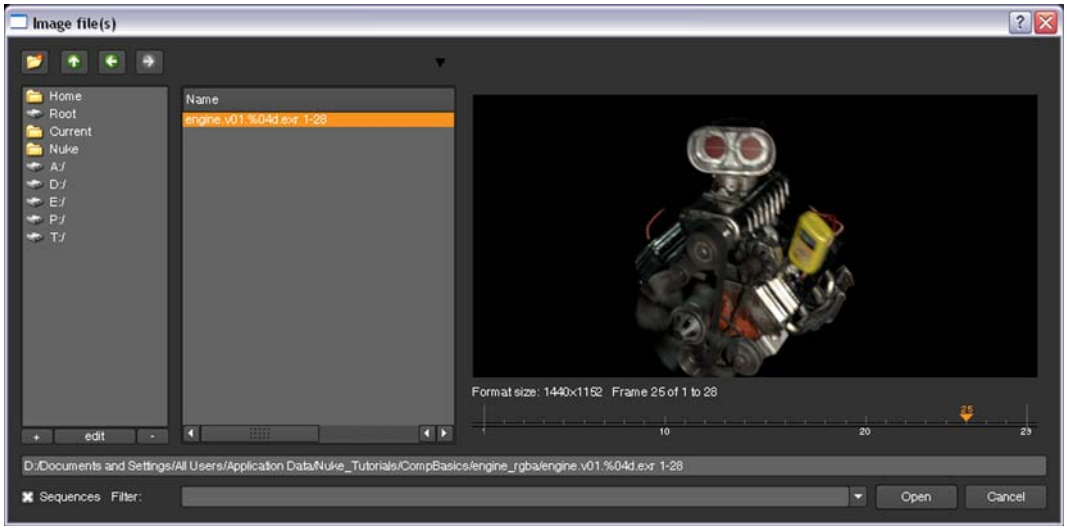
To preview files in the file browser:

1. Click the black arrow in the top right corner of the file browser.



The file browser expands to include a small viewer.

2. Select the file you want to preview. Nuke displays the file in the file browser.



To select multiple files with the file browser:

1. Browse to the folder where you have the files.
2. **Ctrl**+click on all the files you want to open to select them (Mac users **Cmd**+click).
3. Click **Open**.

Nuke opens all the files you selected.

Undoing and Redoing

Nuke generally gives you an undo history that extends back to the first action of the application's current session.

To undo an action in the workspace:

Select **Edit** > **Undo** (or press **Ctrl/Cmd+Z**). Repeat as necessary.

To redo an action in the workspace:

Select **Edit** > **Redo** (or press **Ctrl/Cmd+Y**). Repeat as necessary.

To undo an change in a properties panel:

Click the **Undo** arrow button in the properties panel.

To redo an change in a properties panel:

Click the **Redo** arrow button in the properties panel.

To undo all changes made after the properties panel was opened:

Click the Revert button.

OR

Right-click on the properties panel and select **Revert knobs** from the menu that opens.

To set all controls back to their default values:

Right-click on the properties panel and select **Set knobs to default** from the menu that opens.

Customizing the Interface

You may be used to a certain way of working, or simply disagree with some of Nuke's default settings. If this is the case, you'll be happy to know that you can customize Nuke's interface to find just the layouts and settings that work for you. You can then save your preferred settings, layouts, and template scripts for future use.

Interface Layouts

When your scripts grow, you may not be able to fit all the different elements on your display at the same time. Luckily, you can customize the windows and panes, so that accessing the elements you often need becomes as quick and easy as possible.

Customizing Panes

You can resize and split panes to make more room for different elements on the screen.

To resize a pane:

Drag the divider line of the pane into a new location.

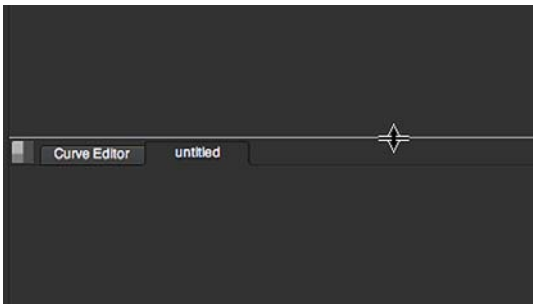


Figure 2-42: Resizing a pane.

To split a pane:

1. Click on the content menu button in the top left corner of the pane.

2. Select **Split Vertical** or **Split Horizontal** from the menu that opens.

Moving the Toolbar

You can move the toolbar into a new position by adding a new panel for it, hiding the panel name and controls, and resizing the panel. For more information on how to do this, see *Adding Tabbed Panels* and *Hiding Tab Names and Controls* below.

Adding Tabbed Panels

When you can't fit more elements to your display, you can use tabs to save space. You can also use tabs to move the toolbar into a new location.

To add tabs:

1. Click on the content menu button in the top left corner of the pane.
2. Select the type of tab you want to add, for example, **Node Toolbar**, **Node Graph**, **New Viewer**, or **Script Editor**.

   The new tab is added on top of the existing tabs.

To move tabs:

Click on the name of the tab and drag the tab to a new position inside the same pane or in another pane.

To close tabs:

1. Make sure you are viewing the tab you want to close.
2. Click the "**x**" button in the top right corner of the current tab.

Tabs and Floating Windows

You can turn tabs and panes into floating windows and vice versa.

To turn a tab or pane into a floating window:

1. Make sure you are viewing the tab or pane you want to float.
2. From the content menu, select **Float Tab** or **Float Pane**.

   Alternatively, in the case of tabs, you can also do one of the following:

   - Click the float button in the top right corner of the current tab.

   OR

   - **Ctrl**+click on the tab name (Mac users **Cmd**+click).

     OR

   - Right-click on the tab name and select **Float Tab**.

To turn a floating window into a tab or pane:

Click on the tab name or pane in the floating window and drag it to where you want it to dock.

To close floating windows:

Click the "**x**" button in the top right corner of the tab or pane.

Customizing Windows

To make a pane expand to the size of the window:

1. With your cursor in the pane.
2. Press **spacebar** quickly. (Pressing and holding the spacebar brings up a context senstive menu for that pane.)

To make a window fullscreen:

1. Make sure the window you want to make fullscreen is active. This could be the main application window or a floating Viewer. Making it fullscreen removes the window borders.
2. Press **Alt+S**.

Hiding Tab Names and Controls

You can hide the names and control buttons of tabs, as you may not need them with all panels, such as the Toolbar panel.

To hide the names and controls on tabs:

From the content menu, disable **Show Tabs**.

To show the names and controls on tabs:

1. Move your mouse pointer over the very top of the pane area until the top edge of the pane brightens up.
2. Right-click to open the content menu.
3. From the content menu, enable **Show Tabs**.

Saving Layouts

You can save up to six favourite layouts and retrieve them as necessary. You can also use these features to setup the Nuke interface for dual monitors.

To save a layout:

1. Open and arrange the Nuke panes, panels, and tabs as desired.
2. Select **Layout** > **Save layout 1 (startup default)** (or press **Ctrl/Cmd+F1**) to save the default layout. Select **Yes** in the confirmation dialog that appears.

   This step shows **F1** as the keystroke, but it can be any key between **F1** and **F6**.

To retrieve a layout:

Select **Layout** > **Restore layout 1(startup default)** (or press **Shift+F1**) to retrieve the default layout.

F1 is used here, but if you saved a layout to a different function key—any key between **F1** and **F6**—then you can press **Shift** followed by the key to retrieve the layout.

To use window layout 1 as the default when loading scripts:

1. Select **Edit** > **Preferences** to open the preferences dialog.
2. Go to the **Windows** tab.
3. Check **use window layout 1 when loading scripts**. This is usually checked by default.

3 Managing Scripts

In this chapter, you learn about Nuke's project files called *scripts*. The topics covered include setting up, saving, and loading scripts.

Working with Multiple Image Formats

Nuke supports multiple file formats, such as Cineon, TIFF, OpenEXR, HDRI, and RAW camera data (via the dcraw command-line program), and allows you to mix them all within the same composite. By default, Nuke converts all imported sequences to its native 32-bit linear RGB colorspace. You can, however, use the Colorspace node to force one of several color models, including sRGB, Cineon, rec709, gamma 1.80/2.20, HSV, or HSL. The Log2Lin node lets you convert between logarithmic and linear colourspace (and vice-versa).

There are no restrictions on image resolution—you can freely mix elements of any resolution within the same script. You can, for example, use a 2k film plate as the background for video shot in PAL format, and then output the result in HD1080i. Nuke automatically adjusts its Viewer to accommodate the image you're viewing.

8-, 16-, and 32-Bit Image Processing

Some digital compositing systems, especially those geared for video work, are optimized for processing exclusively 8-bit elements (that is, images with 256 intensity values per channel). Other systems allow for the mixing of 8, 16, and 32-bit elements.

For Nuke, which began as a film effects tool, image quality is paramount. Thus, it supports the processing of exclusively 32-bit-per channel elements (Elements with lower bit depths are upconverted to 32 bits per channel upon import.) Thirty-two bit support allows for a much richer palette of colours and floating point precision in all script calculations. In practice, this means that Nuke carries out every operation—from an increase in gamma to a transform—with much greater accuracy than a lower-bit-depth system.

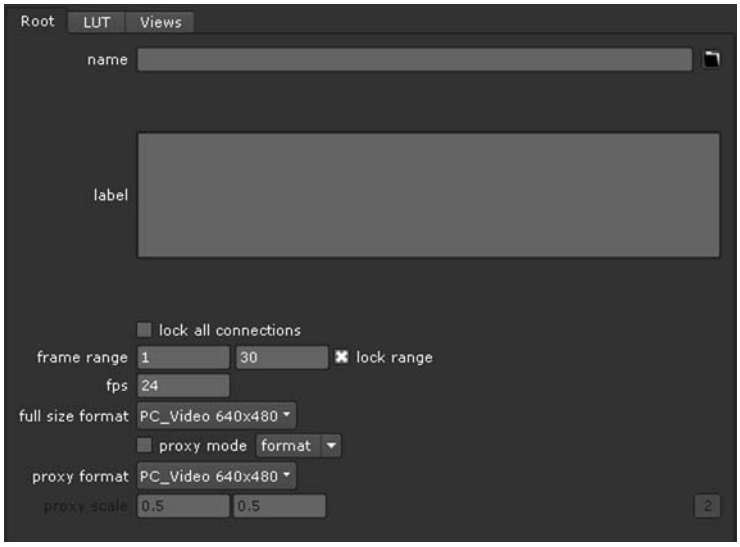
You might expect that this boost in image quality would result in big sacrifice in speed. This is not the case for a couple of reasons. First, Nuke can process scripts with great efficiency because they are uniformly in a 32-bit colorspace. Moreover, hardware manufacturers now routinely design their CPUs for 32-bit processing, making them a efficient match for 32-bit-per-channel images.

Setting Up Your Script

When you start working on a script, you should first define the settings for it. This involves assigning the script a name, frame range, frame rate, and default full and proxy resolution format.

Name, Timespan, and Frame Rate

1. Select **Edit** > **Project settings**, or simply press **S** over a blank portion of the workspace. The Project settings properties panel appears.



2. On the **Root** tab, type a name for the script (say, **firstcomp.nk**) in the **name** field. Nuke's scripts always have the extension .nk.
3. Type the numbers of the first and last frames in the **frame range** fields to define length of time for your "shot."
4. In the **fps** field, enter the rate in frames per second (fps) at which you want your script's Viewers to play back footage. For film-based elements, 24 fps is appropriate.

Full-size Formats

When you start a new project in Nuke, you need to set up a full-size format for the project. The full-size format determines the size of the "black" image that you get from any disconnected node inputs. It also sets the default size of any script-generated elements, such as Constants, Colorbars, and Bezier shapes.

Note

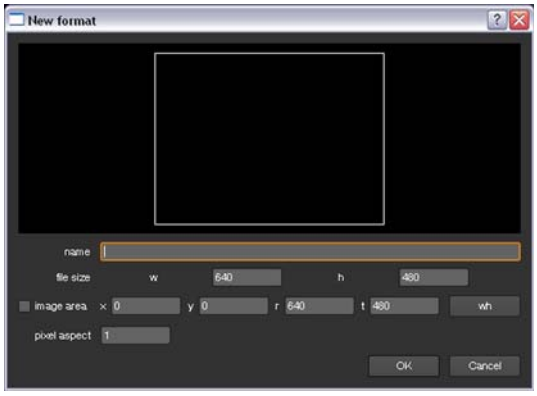
The full-size format does not affect the format of the elements you read into your script. Nuke is resolution-independent, which means it will respect and keep the resolution of your images. It won't automatically crop or pad elements to match the project settings.

If you want elements you read in to conform to the project settings, you can do this manually using the Reformat node. For more information, see *Reformatting Image Sequences* on page 99.

The full-size format is also used to calculate proxy scaling if a proxy format is used. For more information on the proxy mode and proxy formats, see *Proxy Mode* on page 92.

To set up a full-size format:

1. If it's not already open, select **Edit** > **Project settings** (or press **S**) to display the Project settings dialog.
2. From the **full size format** pulldown menu, select the resolution for the final output of rendered images. If the format you want to use is not in the list, select **new**. The *New format* dialog appears.



In the **name** field, enter a name for the new format.

In the **file size** fields, define the width and height of the format.

If you like, you can also define additional information, such as offsets and pixel aspect ratio.

Click **OK** to save the format. It now appears in the pulldown menu where you can select it.

Proxy Mode

When compositing with Nuke, you can work in two different modes: the *full-size mode* or *proxy mode*. In the full-size mode, images are read in exactly as they are on the disk, and all positions are in actual pixels in these images. This is the mode you want to use for accurate feedback and when rendering the final output.

The proxy mode, instead, lets you work with a low-res version of the final output to speed up rendering and display calculation. This can be a down-scaled image that Nuke generates on the fly. Alternatively, you can use Read nodes to specify a proxy file that is read in place of full-size images (for more information on how to do this, see *Loading Image Sequences* on page 97).

The proxy settings you define in the project settings affect both the proxies Nuke generates on the fly and proxies read from files. Below, we discuss setting a proxy format and/or a proxy scale.

Note

Note that proxy versions of images are only used if you have activated the proxy mode (see *Toggling In and Out of Proxy Mode* on page 95). When the proxy mode is off, Nuke will always use the full-res files.

Proxy Format and Proxy Scale

In the Project settings dialog, you have the option of defining a *proxy format* and/or a *proxy scale* that you use in the proxy mode.

For the proxy *format*, you can define the image resolution as well as additional information about offsets and pixel aspect ratio. When using the proxy format, the scaling is proportionate to the full-size/proxy format relationship (not scaled to the proxy format).

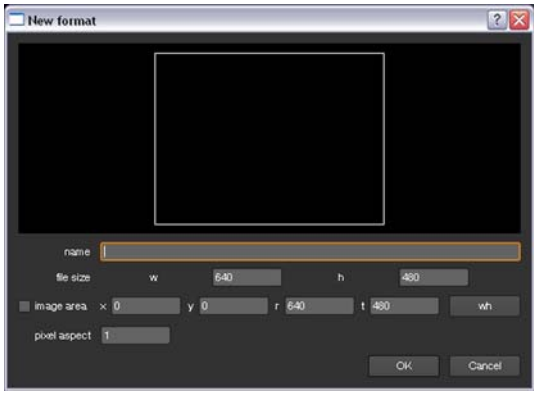
For the proxy *scale*, you only define a simple scale factor by which your images are scaled down whenever the proxy mode is activated. For example, you can use the scale factor of 0.5 to scale your images down to half the size.

If you like, you can define both a proxy format and a proxy scale, and then choose which one to use in proxy mode. A proxy scale is easier to set up, but a proxy format gives you more control over the low-res versions of your images. Below, we first describe how to set up proxy formats and then how to define a proxy scale.

To set up proxy *formats*:

1. If it's not already open, select **Edit** > **Project settings** (or press **S**) to display the Project settings properties panel.
2. If you want to use a proxy format (rather than a proxy scale) whenever the proxy mode is activated, select **format** from the pulldown menu on the right.
3. From the **proxy format** menu, select the resolution to use while working to speed things up. Notice that your images are not scaled to this resolution, but the scaling is proportionate to the full-size/proxy format relationship. Nuke divides the proxy format width by the full-size width and uses the result as the scale factor.

   If the proxy format you want to use is not in the list, select **new**.The *New format* dialog appears.



   In the **name** field, enter a name for the new format.

   In the **file size** fields, define the width and height of the format.

Tip

You can type formulas in numeric fields to do quick calculations. For example, if your full-size format width is 4096 and you want your proxy format width to be 1/2 of that, you can enter **4096/2** in the **New format** dialog's **file size w** field and press **Enter**. Nuke then calculates the new width for you.

   Click **OK** to save the format. It now appears in the pulldown menu where you can select it.

4. To activate the proxy mode and use the low-res format for calculations and display, check **proxy mode**.

Alternatively, you can use the proxy toggle in the Viewer controls, or press **Ctrl+P** (**Cmd+P** on a Mac). For more information, see *Toggling In and Out of Proxy Mode* on page 95.

To setup a proxy scale:

1. If it's not already open, select **Edit** > **Project settings** (or press **S**) to display the Project settings properties panel.
2. Select **scale** from the pulldown menu in the Project settings dialog.
3. Using the **proxy scale** input field or slider, specify the factor by which you want to scale the width and height of your images. For example, if you want to scale them down by 50%, use the value of **0.5**.
4. To activate the proxy mode and use the low-res format for calculations and display, check **proxy mode**.

Alternatively, you can use the proxy toggle in the Viewer controls, or press **Ctrl+P** (**Cmd+P** on a Mac). For more information, see *Toggling In and Out of Proxy Mode* on page 95.

Read Nodes and Proxy Files

As an alternative to letting Nuke generate proxies on the fly, proxy files can be specified using a second filename in the Read nodes (for how to do this, see *Loading Image Sequences* on page 97). If you don't have a proxy file, you can create one by activating the proxy mode and rendering your full-size images using a Write node (see *Rendering Output* on page 351).

The proxy file does not need to match the proxy resolution in use. The closest larger image (full-size or proxy file) will be scaled down to the required size. However, if your proxy images match your typical proxy settings, you will save this time.

Write Nodes and Proxy Files

It is worth mentioning here that when a script is rendered in proxy mode, processing will be done at the proxy scale and image output will go to the file name in the Write node's **proxy** field. If you do not specify a proxy file name, the render will fail with an error. It never resizes the proxy image, and it will not write the proxy image over the full-size one.

For more information, see *Rendering Output* on page 351.

Toggling In and Out of Proxy Mode

It's usually smart to work in proxy mode because most operations work quickly and more efficiently under the low-res display. You can switch between low- and high-resolution when you need greater precision (for example, when pulling a key or tracking), or when you're ready for final rendering.

To toggle between full resolution and proxy mode:

1. Click on an empty area of the Nuke window.
2. Press **Ctrl**+**P** to toggle the display mode (**Cmd**+**P** on a Mac).

Nuke automatically scales script elements—bezier shapes, paint curves, garbage masks, tracking curves, and so on—to keep the original placement on the image.

Caching

To ensure fast playback Nuke caches any output to the Viewer to either RAM or disk.

To define the settings for caching:

1. Select **Edit** > **Preferences**.

   The Preferences dialog opens.
2. Under **disk cache**, specify where you want Nuke to cache out data to disk. Pick a local disk (for example, **c:/temp**), preferably with the fastest access time available.
3. From the **disk cache size** menu, pick the maximum size the disk cache can reach. Ensure there is enough space on the disk for this to be reached.

Saving Scripts and Recovering Backups

You know the mantra: save and save often. Nuke provides three ways to save your scripts, plus an automatic timed backup.

Saving Scripts

There are three ways of saving scripts:

- To save a new script, select **File** > **Save as** (or press **Shift**+**Ctrl/Cmd**+**S**).
- To update changes to a script already saved, **File** > **Save** (or press **Ctrl/Cmd**+**S**).
- To save and upgrade to the next version, **File** > **Save new version** (or press **Alt**+**Shift**+**S**).

To save a script:

1. Select **File** > **Save as**.

   The *Save script as* dialog opens.
2. Browse to the directory where you want to store the script. For instructions on using the file browser, see *Using the File Browser* on page 83.
3. In the field in the bottom of the dialog, enter a name for the script after the folder path, for example **firstscript_v01.nk**.

4. Click **Save**.

Tip

The **_v01** string in the end of a script name allows you to use the **Save new version** feature. Selecting **File** > **Save new version** saves the current version of your script and increments its name (that is, saves the different versions under different names using _v01, _v02, _v03, and so on, in the end of file names). This only works when the file name includes a number that can be incremented.

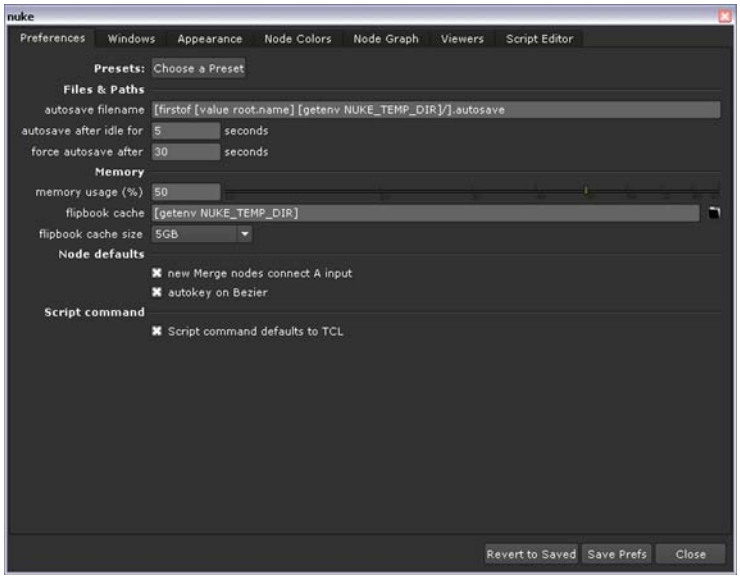
Automatic Backup of Scripts

You can define where and how often Nuke makes automatic backups your files, or turn off the autosave function.

To define autosave options for a script:

1. Select **Edit > Preferences**.

   The Preferences dialog opens.



2. Edit the following settings:
   - **autosave filename** to define where and under what name Nuke saves your automatic backup files. By default, the files are saved in the same folder as your project files with the extension **.autosave**. To change this, enter a full directory pathname in the **autosave filename** field.
   - **autosave after idle for** to define how long (in seconds) Nuke waits before performing an automatic backup after you have left the system idle.
   - **force autosave after** to define how long (in seconds) Nuke waits before performing an automatic backup regardless of whether the system is idle.

3. Click **Save**.

Note

If you close the Preferences dialog without clicking **Save Prefs**, your changes will only affect the current session of Nuke.

Note

For the automatic backup to work, you must save your script first so that the autosave can reference the file. We'd hate for you to lose your work, so please do this early on in the process!

To turn off automatic backup:

1. Select **Edit > Preferences**.

   The Preferences dialog opens.

2. Set the **autosave after idle for** and **force autosave after** fields to **0**.

From now on, Nuke will not perform any automatic backups, and you are more likely to lose your work in the case of a system or power failure.

Recovering Backups

After experiencing a system or power failure, you are likely to want to recover the backup files created by Nuke's autosave function.

To recover backups:

1. Relaunch Nuke.

   A dialog opens that asks you if you want to recover the autosave file.

2. Click **OK**.

Nuke opens the backup file for your use.

There may be times when you don't want to load the autosave file and rather need to load the last saved version. For example, consider a situation where you modified a script, but decided not to commit the changes and so exited Nuke without saving. In all likelihood Nuke autosaved some or all of your changes, in which case if you open the autosaved file you will not be working on the original script, as intended. If you accidentally open an autosaved script, then simply close it and reload the last saved version.

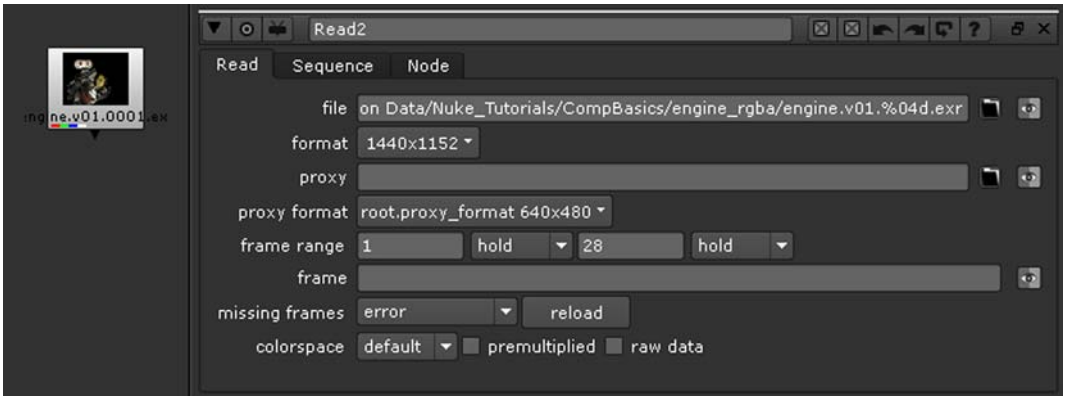
Loading Image Sequences

When you are ready to start compositing, you may want to begin by importing a background or foreground image sequence. Typically, you would read in both full- and proxy-resolution versions of the sequence. You can read in several image sequences in one go.

To import an image sequence:

1. Select **Image** > **Read** (or press **R** over the Nuke window).

A Read node is inserted in the node graph.



Browse to the image sequence you want to import.

For instructions on using the file browser, see *Using the File Browser* on page 83. Select the file you want to open. If you want to open several files at the same time, **Ctrl**+**click** (Mac users **Cmd**+**click**) on the files. Click **Open**.

Nuke imports the image sequence and displays it as a thumbnail on the Read node. Generally, the Read node does not reformat or resize the sequence in any way, and the node's properties panel is updated to display the native resolution and the frame range for the sequence. Note that the **format** and **proxy format** fields in the controls indicate the format of the images, they do not cause the images read from files to be resized to this format.

Note

Nuke reads images from their native format, but the Read node outputs the result using a linear colourspace. If necessary, you can change the Colourspace option in the Read node's properties panel, or insert a **Color** > **Colorspace** node to select the colour scheme you want to output or calculate.

Note

The maximum image size the Nuke Viewer can display is 2^28 = 268,435,456 pixels. This is the same as 16k x 16k, 32k x 8k, 64k x 4k, or 128k x 2k. If your image is larger than this, it will be resized and you will get the following warning:

Viewer image is clipped to <size> x <size>!

For example, if your image resolution is 60,000 x 4473, Nuke is able to display the image because the number of pixels is less than 2^28. However, if the resolution is 60,000 x 4474 (more than 2^28 pixels), the image will be resized to 59998 x 4474.

In addition to the Viewer, this limit is also applied to the bounding box of the images being passed between each node.

2. If you have a proxy version of the image sequence, click the proxy field's folder icon and navigate to the proxy version. Select **Open**. If you don't have a proxy version,

don't worry: Nuke will create one on the fly according to the proxy scale or proxy format settings you specified in the Project settings.

The proxy file does not need to match the proxy resolution in use. The closest larger image (full-size or proxy file) will be scaled down to the required size. However, if your proxy images match your typical proxy scale, you will save this time.

Note

QuickTime .mov files may appear different in Nuke relative to Apple's Final Cut Pro, because Final Cut Pro introduces a gamma compensation based on assumptions about the content of the files and the viewing environment.

Note

If you are using 64-bit Windows, you cannot read QuickTime files into Nuke. This is because Apple has not released QuickTime for 64-bit Windows.

Naming Conventions

The file names of image sequences generally end in a number before the extension, for example image0001.rgb, image0002.rgb, image0003.rgb, and so on. When browsing for files like this, you may notice that the sequence appears as **image%04d.rgb**. Here, **%04d** is Nuke's way of indicating that the number is in a 4-digit incremental format. For a 3-digit format, such as image001.rgb, the frame number variable would be **%03d**.

Nuke's File Browser also understands unpadded file names, such as **image1.rgb**, **image2.rgb**, **image3.rgb**, and so on. They appear as **image%d.rgb**.

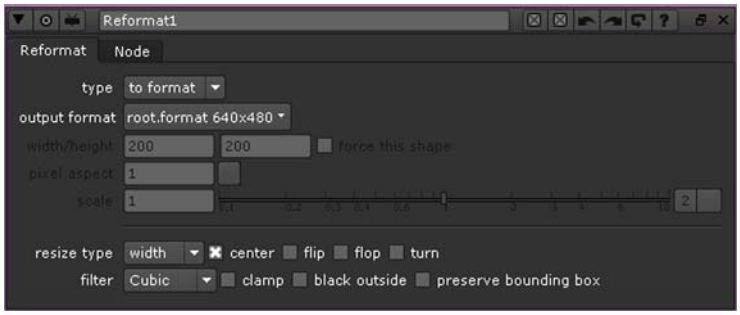
Reformatting Image Sequences

When you import image sequences, Nuke stores their format settings and makes them available to the Reformat node. You can then use the Reformat node to resize and reposition your image sequences to a different format. Reformat nodes also allow you to use plates of varying image resolution on a single script without running into issues when combining them.

To insert a Reformat node:

1. Make sure the Read node you added is currently selected.
2. Select **Transform** > **Reformat**.

   The Reformat node is inserted in the script, and its properties panel opens.



3. From the **output format** pulldown menu, select the format to which you want to output the sequence. If the format does not yet exist, you can select **new** to create a new format from scratch. The default setting, **[root.format],** resizes the image to the format indicated on the Project settings dialog box.
4. You can now use the same Reformat node for any other Read nodes in the script. Simply select the Reformat node and **Edit** > **Copy**. Select another Read node in the script and **Edit** > **Paste**.

Loading Nuke Scripts

When you have built a script and saved it and want to come back to it later, you need to load in an entire script file. You recognise Nuke's script files from the extension **.nk** (for example **firstscript.nk**).

To load a script:

1. Select **File** > **Open** (or press **Ctrl/Cmd+O**).

   The *Script to open* dialog appears.
2. Browse to the script you want to open. For instructions on using the file browser, see *Using the File Browser* on page 83.
3. Click **Open**.

Closing Nuke Scripts

To close a script:

1. Select **File** > **Close** (or press **Ctrl/Cmd+W**).
2. If you have made any unsaved changes to the script, Nuke prompts you to select whether to save them. Click **Yes** to save your changes or **No** to ignore them.

   Nuke is quit and relaunched, as though you ran it again. It does everything it does at start-up, apart from displaying the splash screen. Therefore, you can use **Ctrl/Cmd+W** as a quick way to clear memory, reread plug-in paths, and reload the init.py and menu.py files. (The init.py and menu.py files are files that Nuke runs at start-up and can be used for configuring Nuke. You may want to reload these files if you have made any changes to them.)

Defining Frame Ranges

Several dialogs in Nuke, such as the *Frames to render* and *Frames to flipbook* dialogs, prompt you for a frame range. To define one, you need to enter a starting frame and an ending frame,

separated by a comma. For example, to restrict an action to frames 1, 2, 3, 4, and 5, you would use **1,5** as the frame range.

The following table gives you more examples of frame ranges you can define.

Frame Range	Expands To
3	3
1,10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
1, 10, 2 (frame range from 1 to 10 in steps of 2)	1, 3, 5, 7, 9
1, 10, 3 (frame range from 1 to 10 in steps of 3)	1, 4, 7, 10
10-2/2 (frame range from 10 to 2 in steps of 2)	10, 8, 6, 4, 2
1-4,1 8,10,1 12,14,1 (multiple ranges separated by spaces)	1, 2, 3, 4, 8, 9, 10, 12, 13, 14

Filename Search and Replace

With the Search and Replace function, you can quickly replace all or part of filenames or file-paths in the Read and Write nodes in your script.

To search for a filename or filepath and replace it:

1. Select the Read or Write node(s) where you want to replace all or part of a filename or filepath.
2. Select **Edit** > **Node** > **Filename** > **Search and Replace**.

   OR

   Press **Ctrl+Shift+/** (Mac users press **Cmd+Shift+/**).
3. In the dialog that opens, enter the string you want to search for and the string you want to replace it with. Click **OK**.

Nuke searches for the string in the selected nodes and replaces it with the new string.

Note

You can also enter expressions into the Search and Replace dialog. Just remember that the search field in the dialog only takes regular expressions. Any characters that have specific meanings in regular expressions, such as [and], need to be preceded by the \ character. For example, **[getenv HOME]** would need to be entered as **\[getenv HOME\]**.

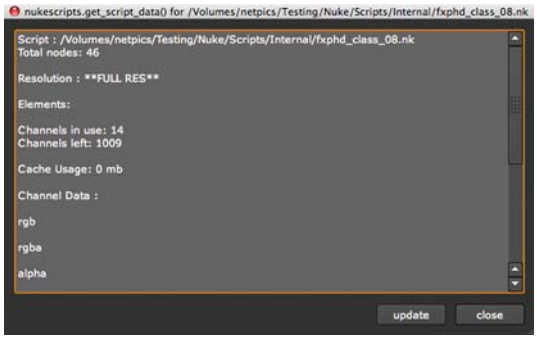
You can also pass flags alongside the expression itself to control how the expression behaves. For example, to perform case-insensitive searches, you can enter **(?i)** in the beginning of the expression or after one or more whitespace characters.

Displaying Script Information

To display script information, such as the node count, channel count, cache usage, and whether the script is in full-resolution or proxy mode, do the following:

1. Select **File** > **Script Info** (or press **Alt+I**).

   The script information window opens.



2. If you make changes to your script while the window is open, click **update** to update the information.
3. To close the information window, click **close**.

Grouping Nodes in the Node Graph

You can group nodes in the Node Graph using the Backdrop node or the Group node. The Backdrop node adds a background box behind the nodes, separating the nodes visually from the rest of the node tree. A Group node, instead, combines a set of nodes into a single node, acting as a nesting container for those nodes.

Grouping Nodes with the Backdrop Node

You can use the Backdrop node to visually group nodes in the Node Graph. Inserting a Backdrop node creates a box behind the nodes. When you move the box, all the nodes that overlap the box are moved, too. By inserting several backdrop nodes, you can group the nodes in your node tree onto boxes of different colours and titles. This makes it easier to find a particular node in a large node tree, for example.

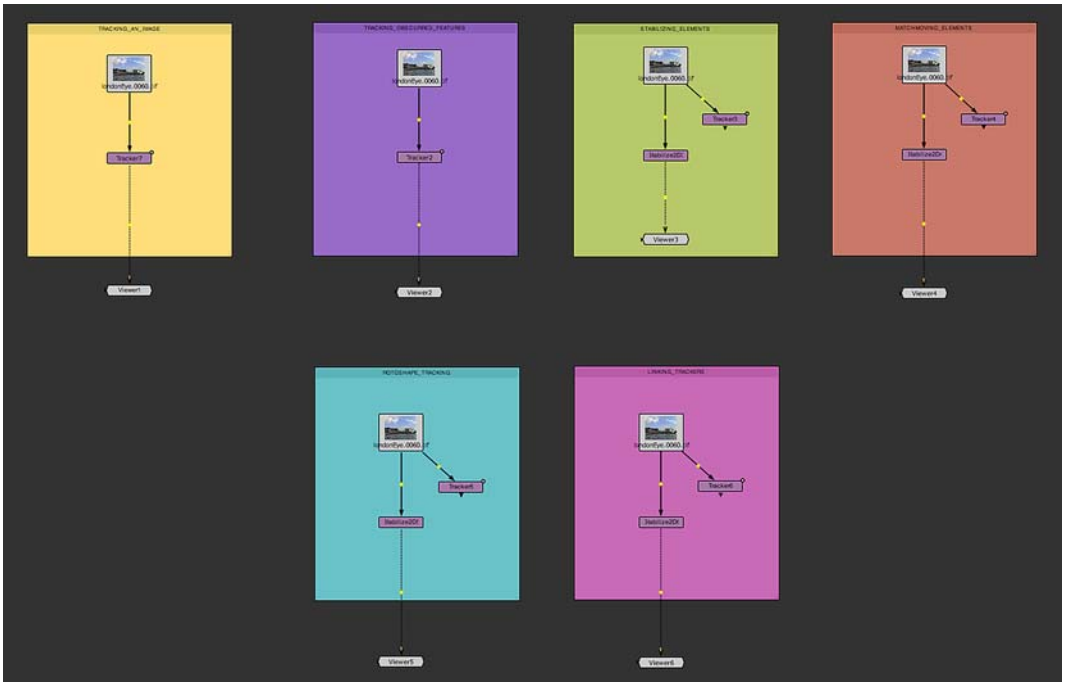
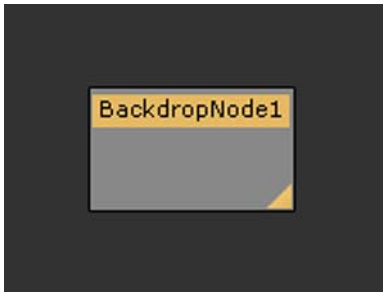


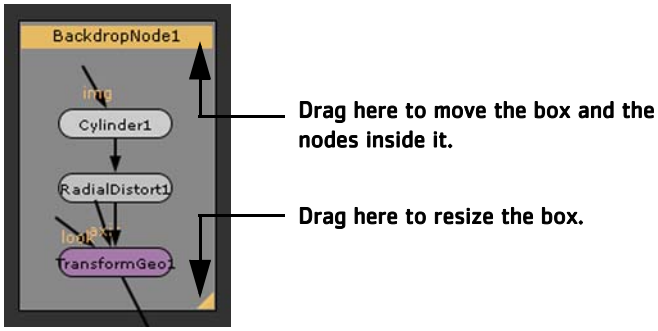
Figure 3-1: Nodes grouped in the Node Graph with Backdrop nodes.

To group nodes with a Backdrop node:

1. Select **Other** > **Backdrop**. A BackdropNode box appears in the Node Graph.

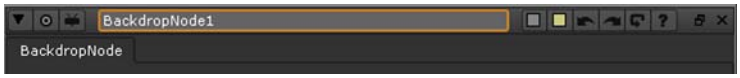


2. Drag the triangle in the lower right corner of the box to resize the box as necessary.
3. Click on the box title bar and drag it to move the box behind the nodes you want to group together. If there are any nodes on the box, they move together with the box.



4. To change the colour of the box, open the Backdrop node's controls by double-clicking on the title bar, then click the left colour button and pick a new colour with the colour picker that appears.

5. To change the title of the box, enter a new name for the Backdrop node in the node's controls.



6. If you later want to remove the box, click on the triangle in the lower right corner to select the box and press **Delete**. Clicking on the title bar and pressing **Delete** removes both the box and the nodes on it.

Grouping Nodes with the Group Node

You can use the Group node to nest multiple nodes inside a single node.

To group nodes with a Group node:

1. Select all the nodes you want to nest inside the Group node.

   Select **Edit** > **Node** > **Group** (or press **Ctrl/Cmd+G** on the Node Graph).

The selected nodes are nested into a group. The original nodes are still in the layout and can be deleted if you like. The internal structure of the Group node is shown on a separate tab that opens.

Tip

As an alternative to **Edit** > **Node** > **Group**, you can also select **Other** > **Group** from the Toolbar or the Node Graph right-click menu.

To view the nodes nested inside a Group node:

In the Group node's controls, click the **S** button (short for Show) in the top right corner.

A new tab that contains the nested nodes opens.

To ungroup nodes:

1. In the Group node's controls, click the **S** button in the top right corner.

   A new tab that contains the nested nodes opens.
2. Copy the nodes from the new tab into your script. If you want to lock the connections between the grouped nodes so that they cannot be accidentally disconnected during the copy-paste operation, check **lock all connections** in the Group node's controls.
3. Delete the unnecessary Group node from your script.

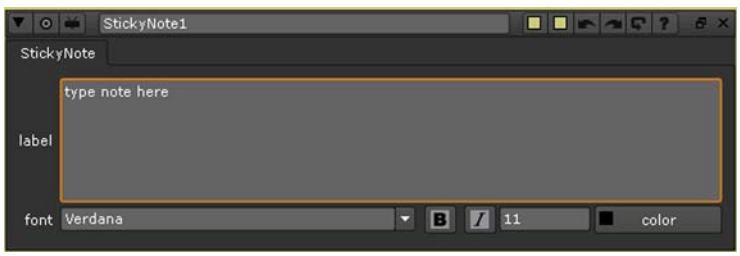
Adding Notes to the Node Graph

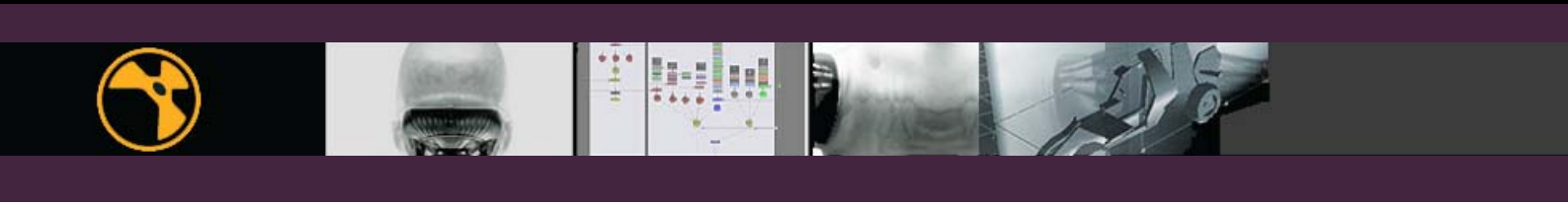
Using the StickyNote node, you can add notes to the Node Graph. The notes can be any text or HTML mark-up. Usually, they are made as annotations to the elements in the node tree.

To add a note to the Node Graph:

1. Click on the part of the Node Graph where you want to add a note.
2. Select **Other** > **StickyNote**. A note box appears in the Node Graph.

3. In the StickyNote controls, enter your note in the **label** field. If you like, you can use HTML mark-up. For example,
   - to have a note appear in bold, you can use **my note.** This would appear as **my note**.
   - to have a note appear in italics, you can use **my note**. This would appear as *my note*.





REFERENCE

Each chapter in this section explains in detail a key feature of Nuke. You can use the section to familiarise yourself with the features you are particularly interested in, or to get answers to specific problems that arise during compositing.

Organisation of the Section

These are the topics covered by this section:

- Chapter 1, *Reformatting Elements*, describes how you can reformat images through scaling, cropping, and pixel aspect adjustments. This chapter also covers working with bounding boxes.
- Chapter 2, *Channels*, shows you how to manage image data using Nuke's unique 1023-channel workflow.
- Chapter 3, *Merging Images*, teaches you how to layer background and foreground elements together, create contact sheets, and copy rectangles from one image to another.
- Chapter 4, *Colour Correction and Colour Space*, explains a broad sampling of Nuke's many colour correction tools.
- Chapter 5, *Transforming Elements*, covers the tools for changing the size, location, and orientation of an image, including how to translate, scale, rotate, and skew elements in 2D and 3D space. This chapter also describes adding motion blur.
- Chapter 6, *Tracking and Stabilising*, shows how to generate and edit 2D tracking data for purposes of removing unwanted motion or applying it to other elements.
- Chapter 7, *Primatte*, teaches you to use the blue/greenscreen keyer Primatte in Nuke.
- Chapter 8, *Using Paint*, describes how to use Nuke's Paint node.
- Chapter 9, *Temporal Operations*, explains how to apply time-based effects like clip retiming and motion blur. This chapter also explains how to perform editorial tasks, such as trimming and slipping.
- Chapter 10, *Warping and Morphing Images*, teaches you to use the GridWarp and SplineWarp nodes to warp and morph images.
- Chapter 11, *Creating Effects*, describes how you can create effects, such as star filter effects, on your images.
- Chapter 12, *Analysing Frame Sequences*, explains how to use the CurveTool node to analyse and match image sequences.
- Chapter 13, *3D Compositing*, teaches you how to create and manipulate 3D scenes composed of objects, materials, lights, and cameras.
- Chapter 14, *Working with Stereoscopic Projects*, describes how to composite stereoscopic material in Nuke.
- Chapter 15, *Previews and Rendering*, teaches you how to write out image sequences from scripts in order to preview results or create final elements.

- Chapter 16, *Expressions*, explains how to apply expressions or scripting commands to Nuke parameters.
- Chapter 17, *Setting Interface Preferences*, discusses the available preference settings that you can use to make behaviour and display adjustments to the interface.
- Chapter 18, *The Script Editor and Python*, teaches you to use Python in Nuke's Script Editor to automate long procedures, for example.
- Chapter 19, *Configuring Nuke*, explains how to set up Nuke for multiple artists working on the same project.

1 Reformatting Elements

This chapter teaches you how to reformat images through scaling, cropping, and pixel aspect adjustments. You will also learn to adjust bounding boxes to minimize processing and rendering times.

Reformatting Images

This section discusses scaling operations with specific regard to reformatting elements to match specific resolutions and pixel aspect ratios. Nuke includes at least two nodes designed for reformatting elements: Reformat, and Crop.

Using the Reformat Node

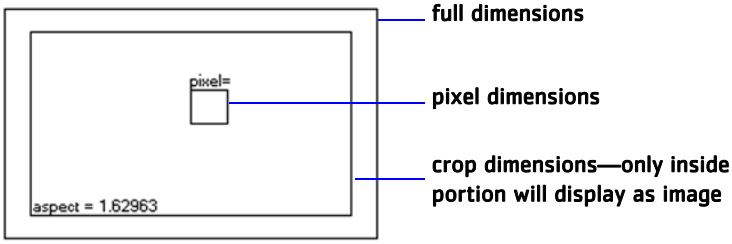
You can use the Reformat node for three different purposes:

1. To generate image sequences that match a desired image format in terms of both resolution and pixel aspect ratio (the width to height ratio of the format's individual pixels).
2. To create postage stamps (low resolution frames which you might post to the web in order to storyboard a sequence). The node scales the frame until it fits inside a rectangle whose dimensions you specify. It also sets pixel aspect ratio to one (square).
3. To scale images. The scale factor will be rounded slightly so that the output image has an integer number of pixels in the direction you choose in the Reformat node's controls.

Converting Images to a Desired Image Format

When you read in elements, Nuke stores their format settings and makes them available to the Reformat node. You can then apply one of the existing formats to your images, or create, edit, and delete formats yourself.

When creating a new format from scratch, you define the overall resolution, the cropped resolution (optional) and the pixel aspect ratio. As you define these parameters, the Reformat operator graphically displays them for you in the manner shown below.



To create a new output format:

1. Click **Transform** > **Reformat** to insert a Reformat node at appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. Select **new** from the **output format** pulldown menu. The *New format* dialog appears.
4. Type a name for the new format in the **name** field.
5. In the **file size** fields, type the full output resolution (in pixels).
6. If you want to crop the full output resolution (for example, to create a letter box):

- Check **image area**.
- Increment the **x** field to define the left boundary of the crop. (The display updates to show you the left boundary of the crop relative to the full-size input.)
- Increment the **y** field to define the bottom boundary of the crop.
- Increment the **r** field to define the right boundary of the crop.
- Increment the **t** field to define the top boundary of the crop.

7. If the destination display device for the image sequence uses nonsquare pixels, type the appropriate pixel aspect ratio in the **pixel aspect** field (for example, if your destination is a digital video display, type **1.1**).

Tip

You can also add formats to Nuke via entries to the **menu.py** file:

1. Open the **menu.py** file (located in same directory as your Nuke executable).

2. Add an entry similar to the following example:

nuke.addFormat ("720 486 0 0 720 486 0.9 NTSC_video")

where the numbers specify, respectively, the format's full horizontal resolution, full vertical resolution, left crop position, bottom crop position, right crop position, top crop position, and pixel aspect ratio; and where the final text string designates the format's name.

3. Save and close the **menu.py** file. The next time you launch Nuke the format will be available for selection from the Project settings dialog, Reformat node properties panel, and elsewhere.

To edit a format:

1. Select the format you wish to edit from the **output format** pulldown list, then let the list close.
2. From the same list, select **edit**. The *Edit format* dialog appears.
3. Edit the **name**, **file size**, **image area**, and **pixel aspect** fields as necessary.
4. Click **OK** to save the changes to the format.

To delete a format:

1. Select the format you wish to delete from the **output format** pulldown list, then let the list close.
2. From the same list, select **delete**. The format is removed from the list.

To apply a format:

1. If necessary, click **Transform** > **Reformat** to insert a Reformat node at appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. From the **type** pulldown menu, select **to format**.
4. Select the format you wish to apply from the **output format** pulldown list.

5. From the **resize type** field to choose the method by which you preserve or override the original aspect ratio. Select:
   - **width** to scale the original until its width matches the format's width. Height is then scaled in such a manner as to preserve the original aspect ratio.
   - **height** to scale the original until its height matches the format's height. Width is then scaled in such a manner as to preserve the original aspect ratio.
   - **fit** to scale the original until its smallest side matches the format's smallest side. The original's longer side is then scaled in such a manner as to preserve original aspect ratio.
   - **fill** to scale the original until its longest side matches the format's longest side. The input's shorter side is then scaled in such a manner as to preserve original aspect ratio.
   - **distort** to scale the original until all its sides match the lengths specified by the format. This options does not preserve the original aspect ratio, so distortions may occur.
6. When cropping the output, check **center** to position the crop area at the centre of the frame.
7. Choose the appropriate filtering algorithm from the **filter** pulldown list (see *Choosing a Filtering Algorithm* on page 160).
8. When scaling an image with Key, Simon, and Rifmen filters, you may see a haloing effect which is caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.

Creating Postage Stamps

1. Click **Transform** > **Reformat** to insert a Reformat node at appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. From the type pulldown menu, select **to box**.
4. In the **width** and **height** fields, type the output dimensions. The units are pixels.
5. Use the **resize type** pulldown menu to choose the method by which you preserve or override the original pixel aspect ratio. Select:
   - **width** to scale the original until its width matches the value in the **width** field. Height is then scaled in such a manner as to preserve the original aspect ratio (this means that the output you specified in **height** may not match the result).
   - **height** to scale the original until its height matches the value in the **height** field. Width is then scaled in such a manner as to preserve the original aspect ratio (this means that the output you specified in **width** may not match the result).
   - **fit** to scale the original until its smallest side matches the corresponding value in **width/height**. The longer side is then scaled in such a manner as to preserve original aspect ratio.
   - **fill** to scale the original until its longest side matches the corresponding value in **width/height**. The smallest side is then scaled in such a manner as to preserve original aspect ratio.

- **distort** to scale the original until its sides match the values in the **width/height** fields. This option does not preserve the original aspect ratio, so distortions may occur.

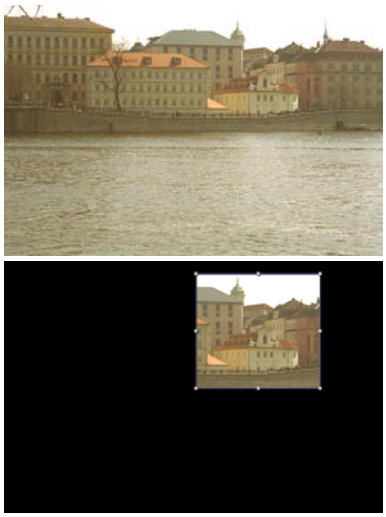
6. Choose the appropriate filtering algorithm from the **filter** pulldown list (see *Choosing a Filtering Algorithm* on page 160).
7. When scaling an image with Key, Simon, and Rifmen filters, you may see a haloing effect which is caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.

Scaling Image Sequences

1. Click **Transform** > **Reformat** to insert a Reformat node at appropriate place in your script (generally before a Write node).
2. Connect a Viewer to the output of the Reformat node so you can see the effect of your changes.
3. From the type pulldown menu, select **scale**.
4. In the **scale** fields, enter scale factors for the width and the height. To scale both directions together using the same scale factor, click the **2** button.
5. Use the **resize type** pulldown menu to choose the method by which you preserve or override the original pixel aspect ratio. Select:
   - **width** to scale the original so that it fills the output width. Height is then scaled in such a manner as to preserve the original aspect ratio.
   - **height** to scale the original so that it fills the output height. Width is then scaled in such a manner as to preserve the original aspect ratio.
   - **fit** to scale the original so that its smallest side fills the output width or height. The longest side is then scaled in such a manner as to preserve the original aspect ratio.
   - **fill** to scale the original so that its longest side fills the output width or height. The smallest side is then scaled in such a manner as to preserve the original aspect ratio.
   - **distort** to scale the original so that both sides fill the output dimensions. This option does not preserve the original aspect ratio, so distortions may occur.
6. Choose the appropriate filtering algorithm from the **filter** pulldown list (see *Choosing a Filtering Algorithm* on page 160).
7. When scaling an image with Key, Simon, and Rifmen filters, you may see a haloing effect which is caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.

Cropping Elements

To *crop* a frame is to cut out the unwanted portions of the image area.



To crop elements:

1. Click **Transform** > **Crop** to insert a Crop node at appropriate place in your script.
2. Connect a Viewer to the output of the Crop node so you can see the effect of your changes.
3. Define the crop boundaries:
   - In the Viewer, drag on any side of the frame to reposition it.

   - Or, in the Crop properties panel, increment or decrement the **box** field (**x** stands for left side; **y**, for bottom side; **r**, for right side; and **t**, for top side).
4. To fill the cropped portion with black, check **black outside**. To fill the cropped portion by expanding the edges of the image, uncheck **black outside**. To adjust the image output format to match the cropped image, check **reformat**.
5. Increment the **softness** field if you wish to vignette the edges of the cropped portion.

Adjusting the Bounding Box

The bounding box defines the area of the frame that Nuke sees as having valid image data. The larger the bounding box is, the longer it takes Nuke to process and render the images. To minimize processing and rendering times, you can crop the bounding box. Occasionally, the bounding box may also be too small, in which case you need to expand it.

To adjust the bounding box, you can use the AdjBBox and CopyBBox nodes. The AdjBBox node lets you both crop and expand the bounding box edges, whereas with the CopyBBox node you can copy a bounding box from one input to another. If needed, you can also add a black outside edge to the bounding box using the BlackOutside node.

Resizing the Bounding Box

The AdjBBox node lets you expand or crop the edges of the bounding box by a specified number of pixels.

An expanded bounding box

A cropped bounding box

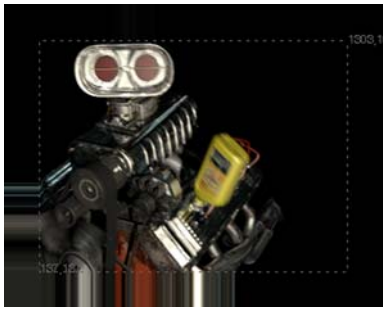
Figure 1-1: Resizing the bounding box with the AdjBBox node.

For example, if you have an image with lots of black (0,0,0,0), you can adjust the bounding box to contain just the useful area so that Nuke won't waste time computing results where there is no change.

To resize the bounding box:

1. Select **Transform** > **AdjustBBox** to insert an AdjBBox node after the image whose bounding box you want to resize.
2. Connect a Viewer to the AdjBBox node, so you can see the effect of your changes.
3. In the AdjBBox controls, adjust the **Add Pixels** slider to increase or decrease the size of the bounding box. By default, 25 pixels are added to the edges of the bounding box.

   Nuke expands or crops the edges of the bounding box. If the bounding box is cropped, whatever is outside the bounding box area gets replicated towards the edges of the image.



Copying a Bounding Box from One Input to Another

Some Nuke operations, such as a merge, can cause an expansion of the bounding box area because Nuke does not know that the extra area is going to be black or another constant colour. Often, you can fix this by copying the bounding box from one of the inputs to the resulting image, thus cutting off this extra area. For this, you can use the CopyBBox node.

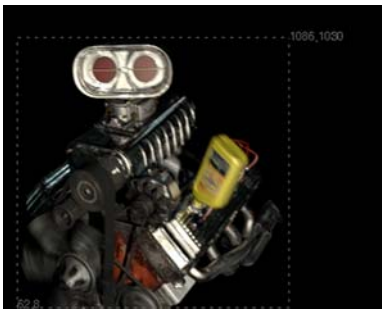
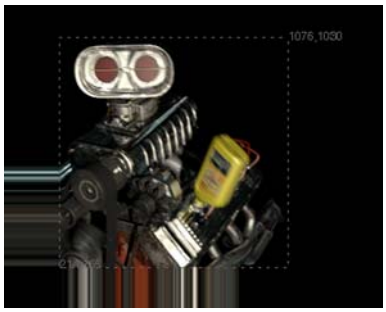


Figure 1-2: A bounding box from one input copied on top of another.

To copy a bounding box:

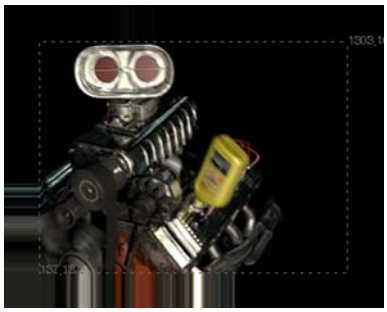
1. Select **Merge** > **CopyBBox** to insert a CopyBBox node after the node whose bounding box you want to use.
2. Connect the image whose bounding box you want to copy to the CopyBBox node's input A, and the image onto which you want to copy the bounding box to input B.

   Nuke copies the bounding box from input A to input B. Whatever is outside the copied bounding box area in image B gets replicated towards the edges of the image.

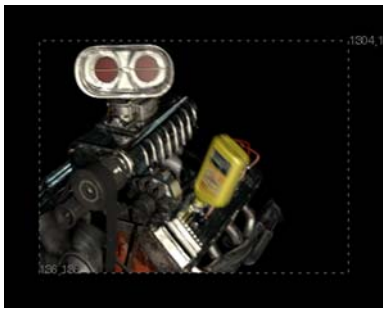


Adding a Black Outside Edge to the Bounding Box

If you adjust a bounding box with the AdjBBox or CopyBBox node, you may notice that whatever is outside the bounding box area gets replicated towards the edges of the image. If necessary, you can remove these replicated edge pixels and fill everything outside the bounding box area with black. To do this, use the BlackOutside node.



A cropped bounding box with replicated edges



The effect of the BlackOutside node

Figure 1-3: Using the BlackOutside node to add a black edge to the bounding box.

To add a black outside edge to the bounding box:

1. Select the image whose edges outside the bounding box you want to fill with black.
2. Choose **Transform** > **BlackOutside** to add a BlackOutside node in an appropriate place in your script.

Nuke fills everything outside the bounding box area with black.

2 Channels

Digital images generally consist of the four standard channels: red, green, blue, and alpha. Nuke allows you to create or import additional channels as masks, lighting passes, and other types of image data. A Nuke script can include up to 1023 uniquely named channels per compositing script.

For example, you can combine multiple render passes from a 3D scene—an image from the red, green, and blue channels, a depth mask (z-depth channel), a shadow pass, a specular pass, lighting passes, and multiple mattes all stored within one image sequence in your composite.

Overview

Before getting into the actual mechanics of channel management, it's important to review some basic concept about how Nuke processes channels.

Understanding Channels

Think of a channel as a container that contains image data. Once created or read into your composite, the image data stored in a channel is available downstream in the network until the value is replaced with something else or the channel is removed. The channel may even be "empty"—depending on where you reference it in the compositing network.

Understanding Channel Sets (Layers)

All channels in a script must exist as part of channel set (also called a *layer*). You're probably familiar with the default channel set—*rgba*—which includes the channels with pixel values of red, green, and blue, and also the alpha channel for transparency.

All channels in a composite must belong to at least one channel set. Some channels, like alpha, may be available in other sets, too. Channel names always include the channel set name as a prefix, like this:
set_name.channel_name.

By default, every script has a channel set called **rgba**. When you first import an image element, Nuke automatically assigns its channels to the rgba set—that is, the image channels are named **rgba.red**, **rgba.blue**, **rgba.green**, and **rgba.alpha**.

The *rgba* set allows for the standard four-channel workflow of most node-based compositing systems. However, you're not limited to these four channels. You can create new channels and assign them to new channel sets up to the limit of 1023 channels per script.

Creating Channels and Channel Sets

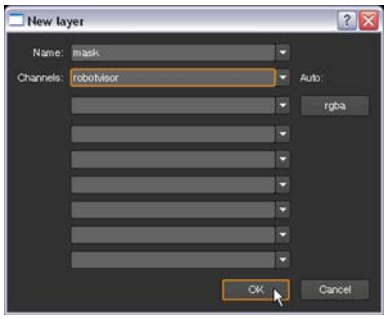
It's important to understand that many types of nodes allow you to direct their output to a specific channel and parent channel set. You have the option of processing these channels in each subsequent node, or leaving them unchanged.

Many nodes feature an **output** or **channels** setting, which lets you direct the output of the current node to a specific channel set and channel. You can also use the **output** or **channels** list to create new channel sets and channels.

Some nodes do not include an **output** or **channels** setting in their parameters. For these, you can connect other nodes, such as Channel Copy or Shuffle, to create and manage channel output in the node tree.

To create a new channel set and/or channel:

1. Open the properties panel for the node whose output will create the new channel.
2. From the **output** or **channels** pulldown list, select **new**.
3. Under **Name**, enter the name of the channel set, and under **Channels** the new channel name.



For example, as shown in the figure above, you would enter **mask** and **robotvisor** to create a new channel ("robotvisor") in the channel set named "mask." If the set does not already exist, Nuke will create it.

4. Click **OK**.

Note

You can also create new channels with the Shuffle and ShuffleCopy nodes. These are explained later, under Swapping Channels.

Calling Channels

By default, most nodes in Nuke attempt to process the current channels in the rgba set and put output in those same channels. However, many nodes also contain an **input** list which lets you select the channels you want to process, and an **output** list to choose the channel(s) where the results should be stored.

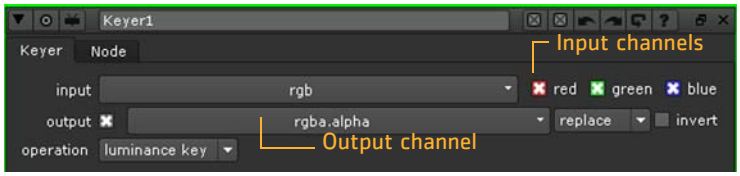


Figure 2-1: Channel input and output

Some nodes also contain **mask** controls and a mask input connector, which let you select a channel for use as a matte to limit operations such as colour corrections. Using these mechanisms, you can point the output of almost any node in the script to any available channel.

The script below attempts to clarify these concepts. Note the script generates six channels (though it could just as well generate 1023). The steps below describe how each channel was created.

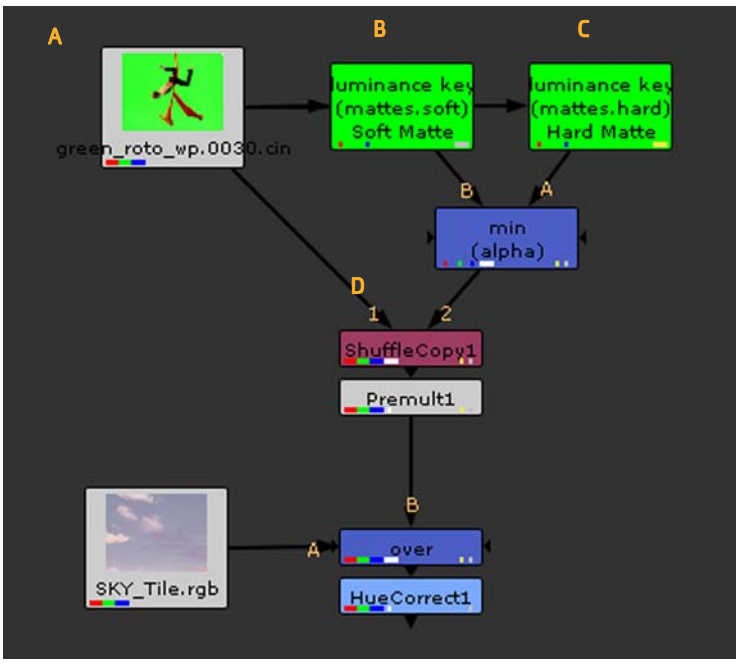


Figure 2-2: A six-channel script.

A. The script reads in the foreground, creating three channels (**red**, **green**, and **blue**), which are by default assigned to the **rgba** set.

   Channel count: 3

B. A low contrast key (**soft**) is pulled and assigned to a new channel set called **mattes**.

   Channel count: 4

C. A high contrast key (**hard**) is pulled and also assigned to the mattes set.

   Channel count: 5

D. The **mattes.hard** and **mattes.soft** channels are mixed to form the final matte (**alpha**), which is assigned to the **rgba** set.

   Channel count: 6

Suppose now that you wanted to perform a colour correction using the output of the Soft Matte node as a mask for the correction. There's no need to pipe the output from that Soft Matte node—it already exists in the data stream along with the other five channels that were created.

You simply attach a colour correction node (for example, the HueCorrect node), then select the appropriate channel from the **mask** controls (in this example, **mattes.soft**). (Again, the **mattes** portion of the name indicates the parent channel set.)

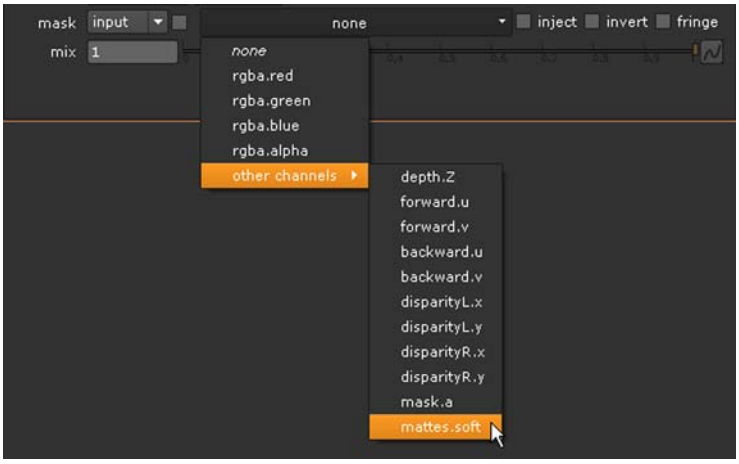


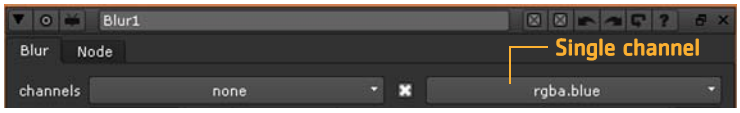
Figure 2-3: Selecting a channel to mask colour correction.

Selecting Input Channels

A node's **channels** field lets you select one or several channels for processing.

To select a single input channel:

1. Open the properties panel of the node into which you wish to feed a channel.
2. From the **channels** field, select **none**.
3. From the right most channel field—the one which typically calls the alpha channel—select the single channel you wish to process.



To select multiple input channels:

1. Open the properties panel of the node into which you wish to feed channels.
2. From the **channels** field, select the channel set containing the channels you wish to process.

   The set's channels appear with check boxes.

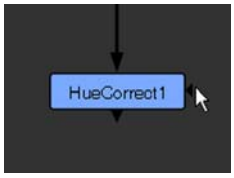
3. Uncheck those channels which you don't wish to process. The node will process all those you leave checked.

Selecting Masks

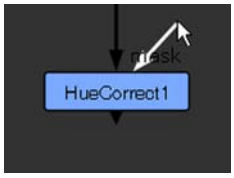
The **mask** controls in a node's properties panel let you select a single channel for use as a matte in a given process (typically, a colour correction). The given process will thereafter be limited to the non-black areas of the selected channel.

You can use one of the script's existing channels as the matte, or attach a mask to the node with a mask input connector.

You can find mask input connectors on colour correction and filter nodes, such as HueCorrect and Blur. At first, they appear as triangles on the right side of the nodes, but when you drag them, they turn into arrows labeled **mask** You connect them the same way as any other connectors. If you cannot see a mask input connector, open the node's properties panel and make sure **mask** is set to **none**.



Before dragging the connector

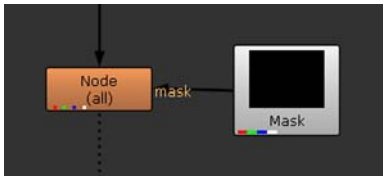


When dragging the connector

Figure 2-4: Using the mask input connectors that appear on some of the colour, filter, channel, and merge nodes.

To select a channel for use as a matte from the mask input:

1. Connect a mask to the node with its mask input connector.



If you cannot see the mask input connector, open the node's controls and make sure **mask** is set to **none**.

By default, when a mask input is connected, the node uses the alpha channel from it as a matte.

2. If you don't want to use the alpha channel as the matte, select the channel you want to use from the **mask** pulldown menu.
3. If you want the mask from the **mask** input copied into the predefined **mask.a** channel, check **inject**. This way, you can use the last mask input again downstream. You can also set a stream of nodes to use **mask.a** as the mask, and then change the masking of all of them by simply connecting a new mask into the mask input connector of the first node.
4. If necessary, check the **invert** box to reverse the mask.

5. If the overall effect of the node is too harsh, you can blend back in some of the input image by adjusting the **mix** slider.

To select a channel for use as a matte from the main input:

1. Make sure nothing is connected to the node's mask input connector. If you disconnect a mask input, the mask input connector disappears, as it is no longer being used.
2. Select the channel you want to use from the **mask** pulldown menu.
3. If necessary, check the **invert** box to reverse the mask.
4. If the overall effect of the node is too harsh, you can blend back in some of the input image by adjusting the **mix** slider.

Tracing Channels

You may have noticed that nodes visually indicate the channels which they are processing (that is, treating in some way) and passing (that is, conveying without any treatment). This is done via a system of coloured rectangles, which allows you to trace the flow of channels throughout a script.

Look closer, for example, at the Over node. The wide rectangles indicate channels which Nuke processes (in this case, the red, green, blue, and alpha channels). The narrow rectangles indicate channels that Nuke passes onto the next node without processing (in this case, the **mattes.soft** and **mattes.hard**).

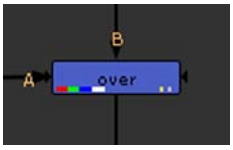


Figure 2-5: Visual confirmation of extra channels.

Renaming Channels

In the course of building your script, you may find it necessary to replace certain channels in a channel set.

To rename a channel:

1. Open the properties panel for a node which has the channel selected on the **channels**, **input**, or **output** pulldown list.
2. Click on the list where the channel is displayed and choose **rename**.

   The Rename Channel box appears.
3. Enter a new name for the channel and click **OK**.

Removing Channels and Channel Sets

When you are done using a channel set or a channel within a set, you may wish, for the sake of clarity, to remove it so that it is no longer passed to downstream nodes. Note that leaving channels in the stream will not itself cause them to be computed; only channels required are computed.

To remove a channel set or a channel within a set:

1. Click **Channel** > **Remove** to insert a Remove node at the appropriate point in your script.
2. In the Remove properties panel, select the channel set you wish to remove from the channels fields.
3. If you don't wish to remove the entire channel set, uncheck the boxes corresponding to the channels which you still wish to be able to call downstream.
4. Click **OK** to close the properties panel.

   The channel set and/or the channels you removed will no longer be displayed in node parameters downstream from the Remove node.

Note

Removing channel sets and or channels does not free up space for the creation of new channels and sets. Once you create a channel it permanently consumes one of the script's 1023 available channel name slots. You are free, however, to rename channels and/or assign them new outputs.

Swapping Channels

Nuke features two main nodes for channel swapping: Shuffle and Shuffle Copy. Shuffle lets you rearrange the channels from a single image (1 input) and then output the result to the next node in your compositing tree. Shuffle Copy lets you rearrange channels from two images (2 inputs) and output the result. Let's take a look at the ShuffleCopy node first.

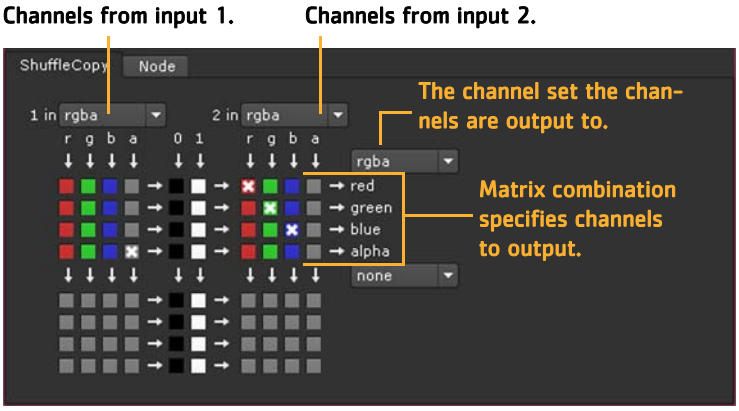


Figure 2-6: The Shuffle Copy matrix.

Channels from Input 1

The first group of channel boxes are the channels supplied by input 1 on the node. As shown above, the foreground element's default **RGBA** set is selected.

Channels from Input 2

The second group of channel boxes are the channels supplied by input 2 on the node.

Channel Outputs

The combination of the boxes checked in the channel matrix create the list of channels that are output to the channel set selected in the top pulldown menu on the right.

This four channel stream acts as *the second set of outputs* from the node. It allows you to output another four channels from the node, for a total of eight channels of output to match the possible eight channels of input.

In this case, this second set of outputs has not been utilized.

Tip

While not required, it's good practice to use the first set of outputs for swapping channels in the current data stream and the second set of outputs for creating new channels. This protects the default **RGBA** set from unintentional overwriting, and makes it easier for other artists to understand the workings of your script.

The basic process then for swapping channels is to first select your incoming channel sets from the **1 in** and **2 in** (or, in the case of the Shuffle node, the **in 1** and **in 2**) pulldown menus. Then, select your outgoing channel sets from the pulldown menus on the right. Then make the actual channel swaps by clicking on the resulting matrix.

For example, to take the simplest case, suppose, you wanted to copy red channel of the RGBA set into its alpha channel. You would click on the matrix to create the following configuration.

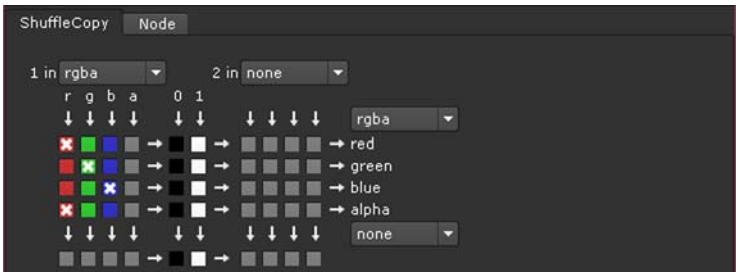


Figure 2-7: Configuring the channel matrix.

You can see that the matrix makes use of the **r** channel (standing for **red**) twice. It goes out once as the red channel, and another time as the **alpha** channel.

Assigning Constants

The shuffle nodes also include parameters that let you assign white (**1**) or black (**0**) constants to any incoming channel. So, for example, to reset the alpha channel to a full-frame image, you would configure the matrix as follows:

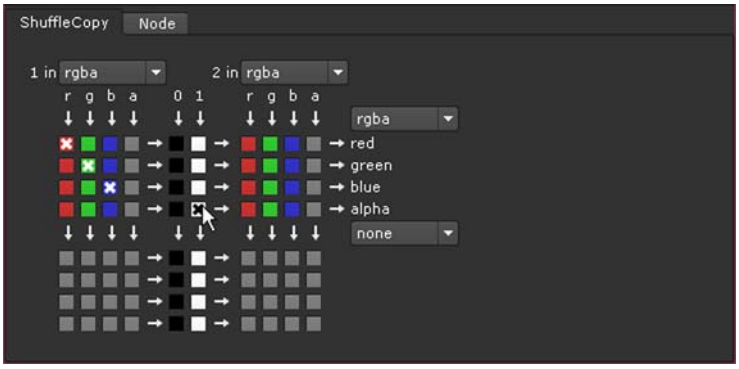


Figure 2-8: Assigning constants to channels.

Creating Swap Channel Sets

Finally, note that if the channel set to which you wish to output channels does not yet exist, you can create it using the **new** option on the pulldown menus on the right. Once you select the **new** option, you follow the same process for creating channel sets as is described in the *Creating Channels and Channel Sets* on page 117.

In Summary

The steps below summarize this discussion on swapping channels.

To swap channels:

1. Click **Channel** > **Shuffle** or **ShuffleCopy** to insert a Shuffle or Shuffle Copy node. Remember you use Shuffle when you only want to swap channels in a single upstream node, and Shuffle copy when you want to swap channels in two separate nodes, like a foreground and background branch.
2. Select the incoming channels from the **In 1** and **In 2** (optional) pull-down lists. You can select up to eight channels in this manner.
3. Select the channel sets to which you wish to direct the incoming channels from the pulldown lists on the right. You can select up to eight channels in this manner.
4. If the outgoing channel set to which you wish to direct channels does not yet exist, create it using the **new** option on the pulldown lists on the right.
5. Click as necessary on the resulting matrix to swap channels.

Tip

If you just need to copy a channel from one data stream into another, use **Channel** > **Copy**, instead of Shuffle Copy. Then, specify the channel to copy and the destination channel for output.

3 Merging Images

With Nuke, you can merge images in a wide variety of ways. In this chapter, we teach you how to use the Merge, ContactSheet, and CopyRectangle nodes. The Merge node is used for layering multiple images together. The ContactSheet node lets you create a contact sheet that shows your frame sequence(s) lined up next to each other in a matrix. The CopyRectangle node can be used for copying a rectangular region from one image to another.

Layering Images Together with the Merge Node

The Merge node with its compositing algorithms allows you to control just how your images are combined.

Note

When using most of the available merge algorithms, Nuke expects premultiplied input images. However, with the **matte** operation you should use unpremultiplied images.

To layer images with the Merge node:

1. Select **Merge** > **Merge** (or press **M** on the node graph) to insert a Merge node after the images you want to layer together.
2. Connect your images to the Merge node's **A** and **B** inputs.
3. If necessary, you can connect multiple A images to the Merge node. Once you have got the A and B inputs connected as instructed in step 2, drag more connectors from the left side of the Merge node to the images you want to use as additional A inputs.

   Each input is merged in the order connected, for example **A1**, **A2**, **A3**, **B**.
4. Connect a Viewer to the output of the Merge node so you can see the effect of your merge operation.
5. In the Merge node's controls, select how you want to layer the images together from the **operation** pulldown menu. The default and the most common operation is **over**, which layers input A over input B according to the alpha of input A. For descriptions of all the available operations, see *Merge Operations* below.
6. Using the **A channels** and **B channels** menus, select which channels to use from the A and B inputs and which channels to use as the A and B alpha. If you want to merge more channels than these and output them into the same channels, select them from the **also merge** pulldown menus and checkboxes.
7. From the **output** menu, select the channels you want to write the merge of the A and B channels to. Channels named in the **also merge** list are written to the same output channels.
8. If necessary, you can also adjust the following controls:
   - To dissolve between the original input B image (at 0) and the full Merge effect (at 1), adjust the **mix** slider. A small light grey square appears on the node in the node graph to indicate that the full effect is not used.
   - If you want to mask the effect of the Merge operation, select the mask channel from the **mask** pulldown menus. To invert the mask, check **invert**. To blur the edges of the mask, check **fringe**.

Note that you should not use the alpha of the inputs for the mask. It produces erroneous results (though the error is often hard to see); you can achieve better results by turning on alpha masking.

- From the **Set BBox to** pulldown menu, select how you want to output the bounding box. The default is **union**, which combines the two bounding boxes. You can also choose **intersection** to set the bounding box to the area where the two bounding boxes overlap, **A** to use the bounding box from input A, or **B** to use the bounding box from input B.
- By default, Nuke assumes that images are in linear colour space. However, if you want to convert colours to the default 8-bit colour space defined in the **LUT** tab of your project settings (usually, sRGB), check **Video colorspace**. The conversion is done before the images are composited together, and the results are converted back to linear afterwards. Any other channels than the red, green, and blue are merged without conversion.

  Checking this option can be useful if you want to duplicate the results you obtained from an application that uses the standard compositing math but applies it to non-linear images (for example, Adobe® Photoshop®). In this case, you typically also need to make sure **premultiplied** is not checked in your Read node controls.
- By default, the same math is applied to the alpha channel as the other channels. However, according to the PDF/SVG specification, many of the merge operations (for example, overlay and hard-light) should set the alpha to (a+b - ab). This way, the input images remain unchanged in the areas where the other image has zero alpha. If you want to enable this, check **alpha masking**.

Merge Operations

When layering images with the Merge node, you need to select a compositing algorithm that determines how the pixel values from one input are calculated with the pixel values from the other to create the new pixel values that are output as the merged image.

The **operation** menu in the Merge node's control panel houses a large number of different compositing algorithms, giving you great flexibility when building your composite. The available algorithms are listed in alphabetical order.

Tip

With many compositing algorithms available, it may sometimes be difficult to find what you're looking for in the **operation** menu. Luckily, there's a quick way of finding a particular operation. With the menu open, you can type a letter to jump to the first operator that starts with that letter. To move to the second operation that starts with the same letter, press the letter again. For example, to select the **screen** operation, open the menu and press **S** twice.

The following table describes each operation and its associated compositing algorithm. You may want to spend some time familiarising yourself with each algorithm in order to be able to determine which operation to use in each situation.

Operation	Algorithm	Description	Illustration	Example Uses
atop	Ab+B(1-a)	Shows the shape of image B, with A covering B where the images overlap.		
average	(A+B)/2	The average of the two images. The result is darker than the original images.		
color-burn	darken B towards A	Image B gets darker based on the luminance of A.		
color-dodge	brighten B towards A	Image B gets brighter based on the luminance of A.		
conjoint-over	A+B(1-a/b), A if a>b	Similar to the over operation, except that if a pixel is partially covered by both a and b, conjoint-over assumes a completely hides b. For instance, two polygons where a and b share some edges but a completely overlaps b. Normal over will produce a slightly transparent seam here.		

Operation	Algorithm	Description	Illustration	Example Uses
copy	A	Only shows image A.		This is useful if you also set the **mix** or **mask** controls so that some of B can still be seen.
difference	abs(A-B)	How much the pixels differ. Also available from **Merge** > **Merges** > **Absminus**.		Useful for comparing two very similar images.
disjoint-over	A+B(1-a)/b, A+B if a+b<1	Similar to the over operation, except that if a pixel is partially covered by both a and b, disjoint-over assumes the two objects do not overlap. For instance, two polygons that touch and share an edge. Normal over will produce a slightly transparent seam here.		This can be useful if you want to merge element a over element b, and element a has element b already held out. For example, you may have a CG character whose hair, skin, and clothing are rendered separately so that each object has the other objects held out of the render. In this case, using the over operation would produce dark lines around the comped objects. This is because over does a hold-out of the background image, meaning the background will be held out twice.
divide	A/B, 0 if A<0 and B<0	Divides the values but stops two negative values from becoming a positive number.		This does not match any photographic operation, but can be used to undo a multiply.

Operation	Algorithm	Description	Illustration	Example Uses
exclusion	A+B-2AB	A more photographic form of difference.		
from	B-A	Image A is subtracted from B.		
geometric	2AB/(A+B)	Another way of averaging two images.		
hard-light	multiply if A<0.5, screen if A>0.5	Image B is lit up by a very bright and sharp light in the shape of image A.		
hypot	sqrt (A*A+B*B)	Resembles the plus and screen operations. The result is not as bright as plus, but brighter than screen. Unlike the screen operation, hypot does not clip pixel values to a maximum of 1.0.		
in	Ab	Only shows the areas of image A that overlap with the alpha of B. Also available from **Merge** > **Merges** > **In**.		Useful for combining mattes.

Operation	Algorithm	Description	Illustration	Example Uses
mask	Ba	This is the reverse of the in operation. Only shows the areas of image B that overlap with the alpha of A.		
matte	Aa+B(1-a)	Premultiplied over. Use unpremultiplied images with this operation. Also available from **Merge** > **Merges** > **Matte**.		
max	max (A,B)	Takes the maximum values of both images. Also available from **Merge** > **Merges** > **Max**.		
min	min (A,B)	Takes the minimum values of both images. Also available from **Merge** > **Merges** > **Min**.		
minus	A-B	Image B is subtracted from A.		
multiply	AB, 0 if A<0 and B<0	Multiplies the values but stops two negative values from becoming a positive number. Also available from **Merge** > **Merges** > **Multiply**.		Used to composite darker values from A with the image of B - dark grey smoke shot against a white background, for example.
out	A(1-b)	Only shows the areas of image A that do not overlap with the alpha of B. Also available from **Merge** > **Merges** > **Out**.		Useful for combining mattes.

Operation	Algorithm	Description	Illustration	Example Uses
over	A+B(1-a)	This is the default operation. Layers image A over B according to the alpha of image A.		This is the most commonly used operation. Used when layering a foreground element over a background plate.
overlay	multiply if B<0.5, screen if B>0.5	Image A brightens image B.		
plus	A+B	The sum of image A and B. Also available from **Merge** > **Merges** > **Plus**. Note that the plus algorithm may result in pixel values higher than 1.0. If you want to clip values to a maximum of 1.0, use the screen operation instead.		Useful for compositing white smoke or fire shot against a dark background.
screen	A+B-AB	Like plus, but clips pixel values to a maximum of 1.0. Also available from **Merge** > **Merges** > **Screen**.		
soft-light		Image B gets lit up. Not as extreme as the hard-light operation.		
stencil	B(1-a)	This is the reverse of the out operation. Only shows the areas of image B that do not overlap with the alpha of A.		

Operation	Algorithm	Description	Illustration	Example Uses
under	A(1-b)+B	This is the reverse of the over operation. Layers image B over A according to the matte of image B.		
xor	A(1-b)+B(1-a)	Shows both image A and B where the images do not overlap.		

Tip

If you have used older versions of Nuke, you may have seen Merge operations called *diagonal*, *nl_over*, and *nl_under*. Diagonal has been renamed and is now called hypot. To get the results of nl_over and nl_under, you can check **Video colorspace** and use over and under.

Generating Contact Sheets

In order to demonstrate, document or manage what you are doing for a project, it can be useful to generate a contact sheet that shows your frame sequence(s) lined up next to each other in a matrix. For this, you can use the ContactSheet node. It generates a contact sheet from all its inputs or from the frames of one input.

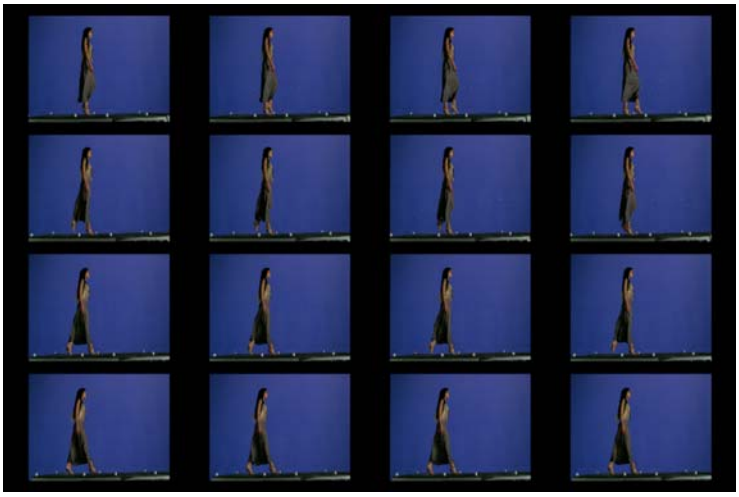
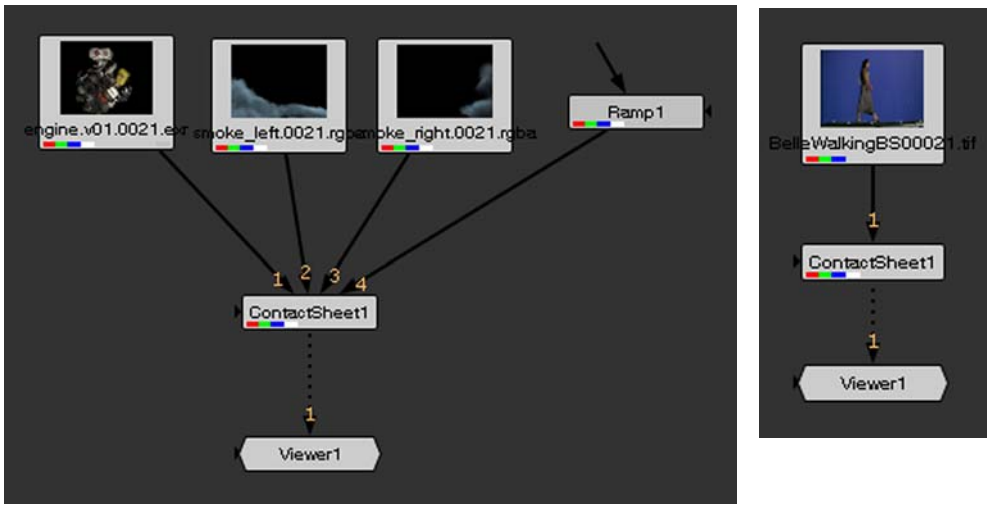


Figure 3-1: A contact sheet generated from the frames of one image sequence.

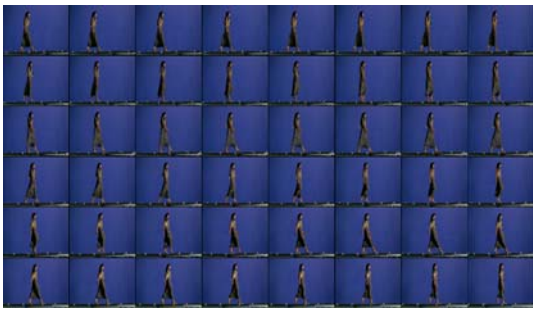
To generate a contact sheet:

1. Select **Merge** > **ContactSheet** to insert a ContactSheet node in your script.

2. Connect the image(s) you want to include in your contact sheet to the numbered input(s) of the ContactSheet node. If you want to include several different image sequences in the contact sheet, use multiple inputs. If you want the contact sheet to include the frames of just one image sequence, use only one input.



3. Connect a Viewer to the ContactSheet node so you can see the effect of your changes.
4. In the ContactSheet properties, define the **Resolution** (width and height) of the entire contact sheet in pixels.
5. If you want to create a contact sheet from the frames of one input, check **Use frames instead of inputs**. In the **Frame Range** field, define the frame range you want to include in the contact sheet.
6. In the **rows/columns** field, specify into how many rows and columns you want to arrange the input images or frames.
7. To adjust the size of the gaps between the images in the contact sheet, increment or decrement the **gap** value.



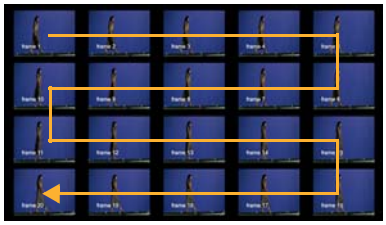
The gap value set to 0.

The gap value set to 50.

8. From the **Row Order** and **Column Order** menus, choose how you want to order the images or frames in the contact sheet:

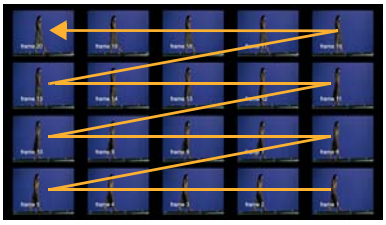
Row Order: TopBottom
Column Order: LeftRight

Row Order: TopBottom
Column Order: RightLeft



Row Order: TopBottom
Column Order: Snake

Row Order: BottomTop
Column Order: LeftRight



Row Order: BottomTop
Column Order: RightLeft

Row Order: BottomTop
Column Order: Snake

Tip

If you want to add any text, such as the frame number, on top of the images in the contact sheet, insert a Text node between the input image(s) and the ContactSheet node.

Copying a Rectangle from One Image to Another

With the CopyRectangle node, you can copy a rectangle from one input on top of another.

Input A

Input B

The output of CopyRectangle

Figure 3-2: Using CopyRectangle to copy a rectangular region from input A onto input B.

The CopyRectangle node can also be used to limit effects, such as colour corrections, to a small region of an image. To do so, you need to use the same image in both input A and B and only perform the colour correction on one input.

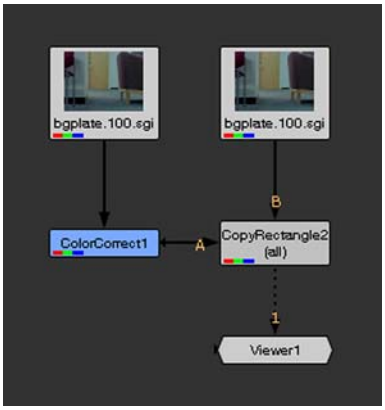


Figure 3-3: A rectangle from input A colour corrected and copied on top of input B.

The original image

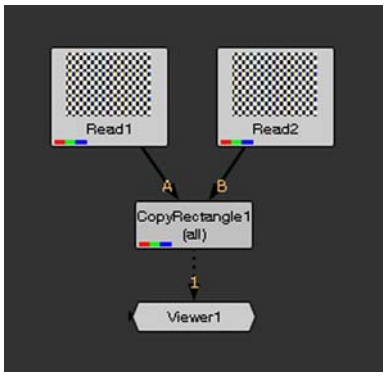
Defining a rectangle with Copy-Rectangle

The colour corrected rectangle on top of the original image

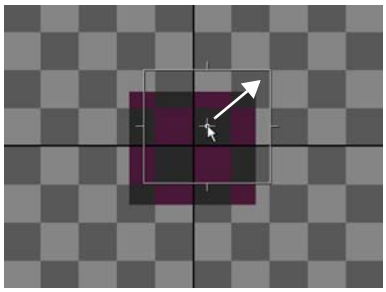
Figure 3-4: Using the CopyRectangle node to limit a colour correction to a rectangular region.

To copy a rectangle from one image to another:

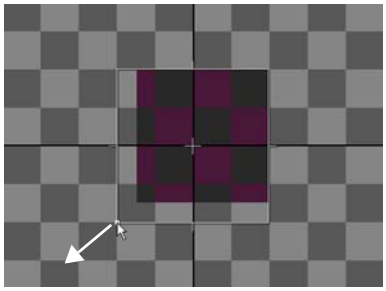
1. Select **Merge** > **CopyRectangle** to insert a CopyRectangle node after the image that has a region you want to copy (input A) and the image you want to copy the region to (input B). Create the following setup:



2. In the CopyRectangle controls, use the **channels** pulldown menu to select the channels you want to copy from input A.
3. To define the rectangle you want to copy, resize and reposition the CopyRectangle overlay in the Viewer. Drag the centre of the overlay to reposition, and the edges to resize. If you cannot see the overlay in the Viewer, open the CopyRectangle properties panel and double-click on the Viewer node in the Node Graph.

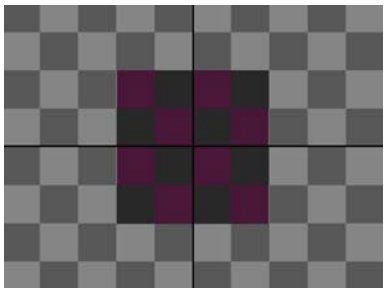


repositioning the rectangle

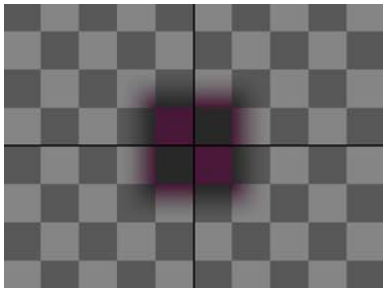


resizing the rectangle

4. To control how soft the edges of the rectangle seem, adjust the **softness** slider. The higher the value, the softer the edges.



low softness value



high softness value

5. To dissolve between the full CopyRectangle effect and input B, adjust the **mix** slider.

4 Colour Correction and Colour Space

This chapter explains how to use Nuke's colour correction nodes to adjust the appearance of the images in your composites. Specifically, you'll learn how to:

- Make tonal adjustments.
- Make basic contrast, gain, gamma, and offset adjustments.
- Make hue, saturation, and value adjustments.
- Apply masks to colour corrections.
- Convert elements into nonnative colour spaces.
- Apply grain.

These topics provide a good overview of Nuke's colour-correction nodes; however not all options are covered here. Look to Nuke's online help for instructions on using the other nodes found under the Color icon in the toolbar.

Making Tonal Adjustments

Defining tonal range (the blackpoint, whitepoint, and neutral value) is typically the first step in colour correcting a clip. Tonal range adjustments often improve contrast, but more importantly, they set the stage for subsequent colour corrections by properly dividing the colourspace into shadow, midtone, and highlight regions.

Figure 4-1: Before and after tonal adjustment.

Several of Nuke's colour correction effects offer tonal adjustment tools. Of these, Grade and Histogram are probably the most intuitive to operate.

Using Histograms

The properties panel for the Histogram node includes a window that graphs the number of pixels at each brightness level. This is a useful gauge to see whether an image has a good distribution of shadows, midtones, and highlights.

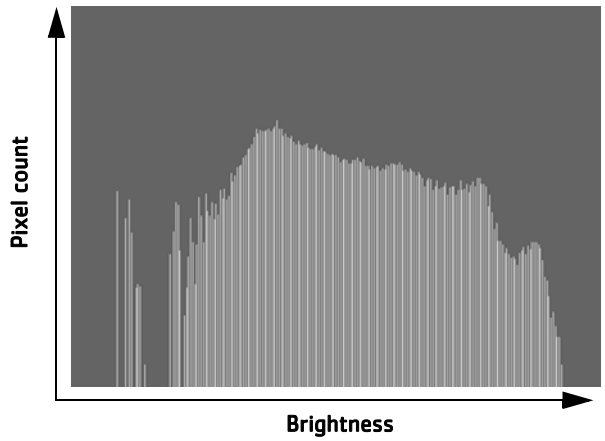


Figure 4-2: The histogram maps the distribution of shadows, midtones, and highlights.

To define tonal range with the Histogram node:

1. Click **Color** > **Histogram** to insert a Histogram node at the appropriate place in your script.
2. Connect a Viewer to the output of the Histogram node so you can see the effect of your changes.
3. Drag the leftmost **input range** slider till it roughly lines up with the initial boundary of the histogram.
4. Drag the rightmost **input range** slider till it roughly lines up with the final boundary of the histogram.
5. Drag the middle input range slider to define the midtone, or neutral, value.

Sampling White and Black points

The Grade node lets you define white and black points by sampling pixels from a Viewer frame.

To define tonal range with Grade:

1. Click **Color** > **Grade** to insert a Grade node at the appropriate place in your script.
2. Connect a Viewer to the output of the Grade node so you can see the effect of your changes.
3. In the Grade properties panel, use the **channels** pulldown list to select the channels you wish to process.
4. Click the **blackpoint** parameter's colour swatch.

   The eye dropper icon appears.
5. In the Viewer, press **Ctrl/Cmd+Shift** while clicking on the pixel you want to define as the blackpoint (typically the darkest pixel).
6. Click the **whitepoint** parameter's colour swatch. The eye dropper icon appears.

7. In the Viewer, press **Ctrl/Cmd+Shift** while clicking on the pixel you want to define as the white point (typically the lightest pixel).

Making Basic Corrections

Adjustments to contrast, gamma, gain, and offset often comprise the bulk of the work in colour correction. Some artists prefer to makes these adjustments via sliders; others prefer curves (which represent the range of colour values in an image.) Nuke's ColorCorrect and ColorLookup nodes offer tools to suit either preference.

Figure 4-3: Contrast boost

Figure 4-4: Gain boost

Figure 4-5: Gamma boost

Figure 4-6: Offset boost

Using Sliders

The ColorCorrect node is particularly convenient for making quick adjustments to contrast, gamma, gain, and offset. A single window houses sliders for all these basic corrections and allows you to apply these to a clip's master (entire tonal range), shadows, midtones, or highlights.

To adjust contrast, gain, gamma or offset with the ColorCorrect node:

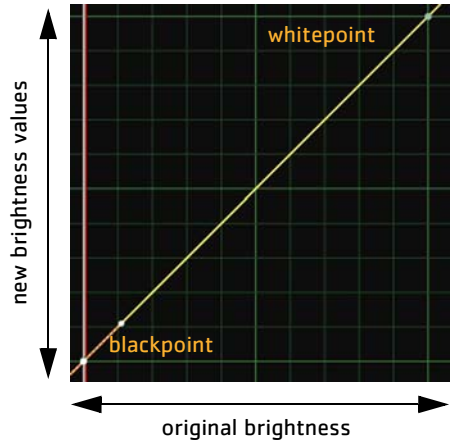
1. Click **Color** > **ColorCorrect** (or press **C**) to insert a ColorCorrect node at the appropriate place in your script.
2. Connect a Viewer to the output of the ColorCorrect node so you can see the effect of your changes.
3. In the ColorCorrect properties panel, use the **channels** pull-down list to select the channels you wish to process.
4. Drag the slider appropriate to the region you want to affect an operation you want to apply.

   For example, to brighten the images highlights, you would drag on the **highlights gain** slider.

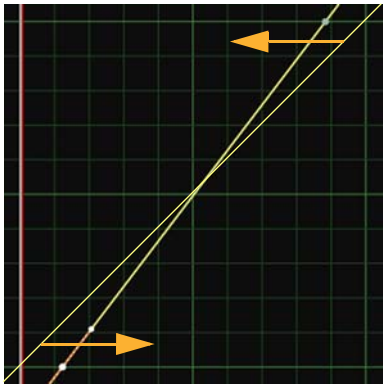
5. Remember too that you can use the colour sliders to apply any of the corrections on a per channel basis.

Using Colour Curves

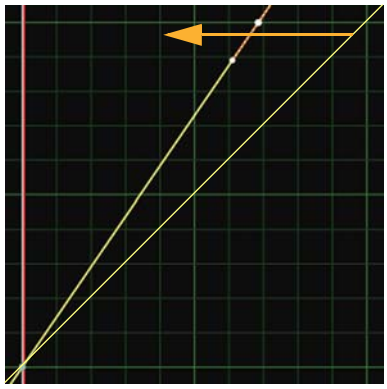
If you prefer to work with colour curves, you can use the ColorLookup node to make contrast, gamma, gain, and offset adjustments (and, in fact, many others). *Colour curves* refer to line graphs of a given colour channel's brightness. The horizontal axis represents the channel's original, or input, values, and the vertical axis represents the channel's new, or output values.



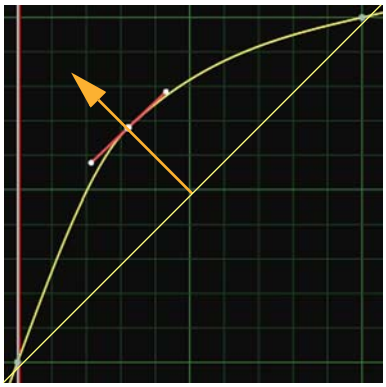
As Figure 4-7 shows, you can edit the ColorLookup node's colour curves to make all of the types of corrections that are possible through the ColorCorrect node—and you can generally make these corrections with more flexibility and precision than is possible with sliders



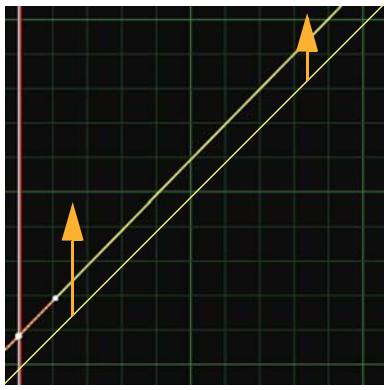
Contrast boost



Gain boost



Gamma boost



Offset boost

Figure 4-7: Corrections through the ColorLookup node's colour curves.

To make basic corrections with the ColorLookup node:

1. Click **Color** > **ColorLookup** to insert a ColorLookup node at the appropriate place in your script.
2. Connect a Viewer to the output of the ColorLookup node so you can see the effect of your changes.
3. In the ColorLookup properties panel, click **red**, **green**, **blue**, or **alpha** if you want to limit the subsequent operations to a particular channel.

   You can select multiple curves in order to edit one curve with reference to another). Otherwise, select the **master** curve (which represents all channels).
4. In the Viewer, drag the cursor over the pixels you want to sample for the correction.

5. In the ColorLookup properties panel, press **Ctrl**+**Alt** (**Cmd**+**Alt** on a Mac) while clicking on the curve to set points at the places where the red, green, and blue lines intersect with the colour curve.

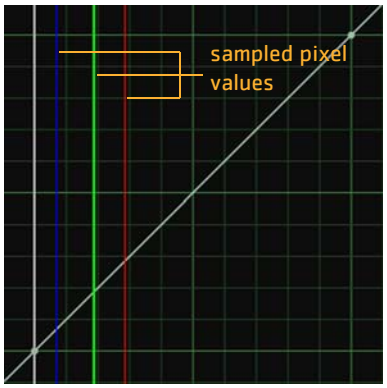


Figure 4-8: Viewing values from a sampled colour.

6. Edit the position of the points and adjust the tangent handles to adjust the curve shape for the colour correction.

Making Hue, Saturation, and Value Adjustments

For certain colour correction tasks like spill suppression, you ideally want to influence only a very narrow range of colour values. For such tasks, it's often helpful to use effects that employ the Hue, Saturation, and Value (HSV) colour model. As its name indicates, the HSV colour model breaks colour into three components:

- *Hue*, which refers to the colour's location on the traditional colour wheel.
- *Saturation*, which refers to the extent to which the colour has "soaked up" its hue.
- *Value*, which refers to the brightness of the colour.

Figure 4-9: Hue shift

Figure 4-10: Saturation decrease

Figure 4-11: Value decrease

Nuke offers effects that allow you to correct the hue, saturation, and value components individually or collectively.

Correcting HSV

Nuke's HSVTool node lets you simultaneously adjust hue, saturation, and value components from a single properties panel. It also features a colour replacement tool. The main strength of this node is the precision it offers in limiting corrections to a narrow swath of colours

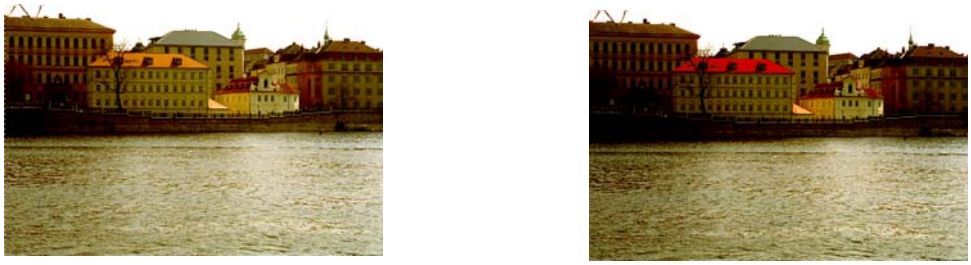


Figure 4-12: Adjusting colour within a specific range of pixel values.

For example, suppose you wanted to add a bit more punch to the waterfront scene by diversifying the rooftop hues. To do so, you could limit the correction to the rooftop's ochre-coloured hues by sampling a few pixels, then shift their values. Because you limited the colour range, the surrounding image would be generally unaffected by the shift.

To make HSV corrections with the HSVTool node:

1. Click **Color** > **HSVTool** to insert an HSVTool node at the appropriate place in your script.

2. Connect a Viewer to the output of the HSVTool node so you can see the effect of your changes.
3. Limit, as appropriate, the range of colours you want subsequent corrections to influence:
    - In the HSVTool properties panel, click the **srccolour** colour swatch. **Ctrl**/**Cmd**+click on the Viewer to sample a single colour displayed, or **Ctrl**/**Cmd**+**Shift**+drag to sample a range of colours. To sample a single colour from the node's input while viewing its output, **Ctrl**/**Cmd**+**Alt**+click on the Viewer. To sample a region from the input, **Ctrl**/**Cmd**+**Alt**+**Shift**+drag on the Viewer.
    - The **Range** sliders on **Hue**, **Saturation**, and **Brightness** clamp to the sampled range.
    - For any colour component, drag on the **Range** sliders to expand the colour range as necessary.
    - For any colour component, drag on the **Range Rolloff** slider to fine tune the colour range. Doing so, adjusts the amount of falloff allowed past the limits defined by the Range sliders.
4. Make the necessary HSV corrections:
    - For hue corrections, drag on the **Rotation** slider to input colour wheel value between 0 and 360.
    - For saturation corrections, drag on the **Saturation Adjustment** slider to input values between **-1** (completely desaturated to some shade of gray) and **1** (completely saturated).
    - For value corrections, drag on the **Brightness Adjustment** slider to input values between **-1** (black) and **1** (white).

You can also make colour replacements using the srccolour and dstcolour parameters: First sample the colour you wish to replace with the srccolour colour swath, then sample the colour which you wish to use as the replacement with the dstcolour colour swath. The colour in dstcolour replaces the colour in srccolour throughout the image.

Also, keep in mind that the HSVTool node makes an excellent keyer. You can use its Hue, Saturation, and Brightness range sliders to precisely select a range of colours, then use the channel output pulldown at the bottom of the dialog to output this selection as a matte channel. This pulldown lets you specify which colour components (hue, saturation, value, etc.) are added to the matte.

Correcting Hue

Nuke's HueCorrect node lets you make precision adjustments to the levels of saturation in a range of hues. You do so via edits to a series of suppression curves.

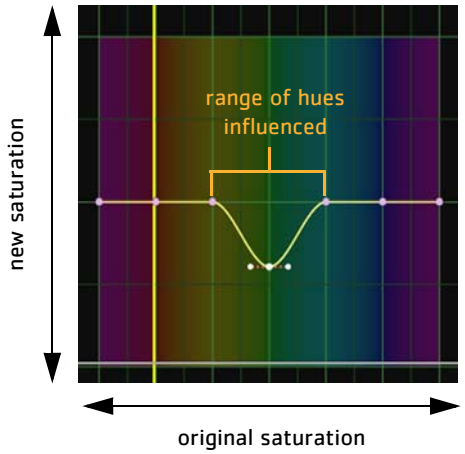


Figure 4-13: Editing the suppression curve.

By choosing which curve you edit and how much of that curve you alter, you can precisely limit the influence of the effect.

Suppressing Spill

For the compositor, HueCorrect is obviously of greatest use in diminishing green, blue, or red-screen spill.

To suppress spill with the HueCorrect node:

1. Click **Color** > **HueCorrect** to insert a node at the appropriate place in your script.
2. Connect a Viewer to the output of the HueCorrect node so you can see the effect of your changes.
3. In the HueCorrect properties panel, choose the channels you want to influence:
   - Click **sat** to influence all channels (red, green, blue, and alpha) equally.
   - Click **lum** to influence all channels, but with luminance weighting in effect (meaning that the red channel receives approximately 30% of the effect; the green, 60%; and the blue, 10%).
   - Click **red** to influence only the red channel.
   - Click **green** to influence only the green channel.
   - Click **blue** to influence only the blue channel.
   - Click **r_sup** to influence all channels but also mix in the values of the original red channel.
   - Click **g_sup** to influence all channels but also mix in the values of the original green channel.

- Click **b_sup** to influence all channels but also mix in the values of the original blue channel.

Note that you can select multiple curves in order to edit one curve with reference to another.

4. If necessary, drag the cursor over the Viewer to sample the image pixels that are representative of the part of the image you want to correct. Then, in the HueCorrect properties panel, press **Ctrl**+**Alt** (**Cmd**+**Alt** on a Mac) while clicking on the curve to plot a particular pixel's value on the curve. This lets you see what portion of the curve you want to edit.
5. Edit the curve as necessary—typically this means dragging down on control points in the hue region that you wish to suppress.

Correcting Saturation

For the times when you just want to correct the saturation component and don't require limiting the correction to any particular channel, you can use Nuke's Saturation node. It's controls are bare bones—basically just a **saturation** slider.

To make saturation corrections with the Saturation node:

1. Click **Color** > **Saturation** to insert a Saturation node at the appropriate place in your script.
2. Connect a Viewer to the output of the Saturation node so you can see the effect of your changes.
3. Drag the **saturation** slider to make the necessary corrections.

Masking Colour Corrections

Virtually all the colour-correction effects in Nuke include **mask** parameters that lets you limit the correction to the non-black pixel values of a matte image. For example, suppose you want to add a blue cast to the following scene without affecting the buildings.

Original image

Colour-corrected image without mask

Mask image

Colour-corrected image limited by mask

Figure 4-14: Masking colour-correction operations.

You could create a garbage mask that covers the river, then boost the red channel's gamma in the area of the frame that underlies the mask.

Typically, mask controls are located toward the bottom of the properties panel. However, in the case of multi-purpose effects like HSVTool, there may be multiple mask controls, so that you can limit each type of correction with a different mask.

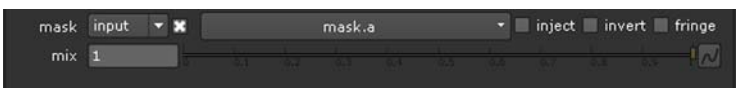


Figure 4-15: Selecting a mask channel.

To mask a colour correction:

1. Open the node's properties panel and locate the **mask** controls.
2. From the first pulldown menu, select:

- **mask** to use a channel from the node's mask input as the matte. Connect the mask to the node with the mask input connector, if you haven't already done so.

  If you check **inject** in the **mask** controls, the mask from the **mask** input is also copied into the predefined **mask.a** channel. This way, you can use the last mask input again downstream. You can also set a stream of nodes to use **mask.a** as the mask, and then change the masking of all of them by simply connecting a new mask into the mask input connector of the first node.

- **input** to use one of the channels in the node's primary input as the matte.
- **output** to use a channel as it exists in the node after the node processes.

3. Select the channel you wish to use as the mask from the pulldown list.
4. If necessary, check the **invert mask** box to reverse the mask.
5. If the overall effect of the node is too harsh, you can blend back in some of the input image by dragging on the **mix** slider.
6. If you want to output only the portion of the frame underlying the mask, check the **(un)premult by** box.

Applying Grain

Grain matching—ensuring that all of the elements in a composite, including those which were digitally generated, look like they were shot on the same film stock—is often one of the final steps in achieving a convincing integration of all of a composite's elements. Nuke offers effects for synthetically creating grain and for reading in practically-created grain (grain derived from actual film stock).

Grainless image

Grained image

Figure 4-16: Here's an example of applying grain to an image.

Using Synthetic Grain

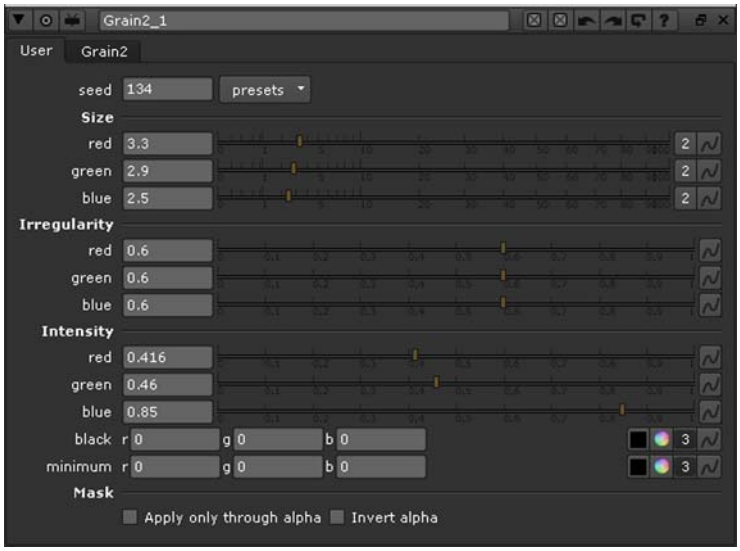
Nuke offers several nodes for creating synthetic grain: Dither, Grain, and ScannedGrain. Of these, Dither is the crudest—it basically just lets you specify the amount of noise per channel.

Grain includes presets for matching film stock and a means for controlling the mix between the generated grain and the backplate. ScannedGrain offers film stock presets, plus synthetic grain controls for applying practical grain.

To add synthetic grain with the Grain node:

1. Click **Draw** > **Grain** to insert a Grain node at the appropriate place in your script.

2. Connect a Viewer to the output of the Grain node so you can see the effect of your changes.



3. From the **presets** pulldown menu, choose one of the film stock you want to match.
4. Adjust the **Size** sliders for the red, green, and blue channels to shrink or enlarge the granules.
5. Adjust the **Irregularity** sliders to increase or decrease the random quality of the grain, according to the different channels.
6. Adjust the **Intensity** sliders to increase or decrease the contrast of the grain against the original image.

Using Practical Grain

Although Nuke's ScannedGrain node offers controls for creating synthetic grain (ones comparable to those just discussed), it's main use is for reading in and applying scanned grain—that is, grain derived from actual film stock. If your facility has such sequences available, you can read them in and apply them using the ScannedGrain node. You can also download grain files from our website for this purpose. Both creating and downloading grain files are described below, as well as using the resulting grain files with the ScannedGrain node.

To create film stock sequences:

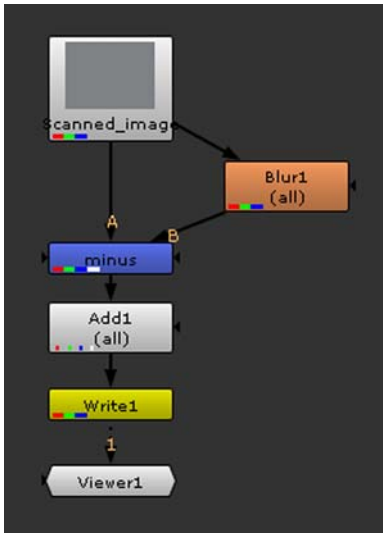
1. Film a gray card. Only about 50 frames are needed.
2. Scan the film in.
3. Select **Image** > **Read** to load the scanned image into Nuke.
4. Add a Blur node (**Filter** > **Blur**) after the image to blur the image until you cannot see any grain. Then, blur the image a bit more.
5. Select **Merge** > **Merge** to insert a Merge node in your script. Connect the A input of the Merge node into the original image, and the B input into the Blur node. Then, open the Merge controls and select **minus** from the **operation** pulldown menu. The blurred image is subtracted from the original image.

The purpose of this and the previous step is to subtract any overall gray level from the grain so that only the grain is left.

6. Select **Color** > **Math** > **Add** to insert an Add node after the minus node. In the Add node controls, enter 0.5 in the **value** field. This adds a value of 0.5 to all channels.

   This step is necessary, because the ScannedGrain node subtracts 0.5 from the channels when it reads the grain file (the subtraction is needed to store negative numbers in most file formats).

7. Select **Image** > **Write** to insert a Write node after the Add node. Render the output. Any file format will do (for example, we have used the .rgb extension in the grain files on our website).

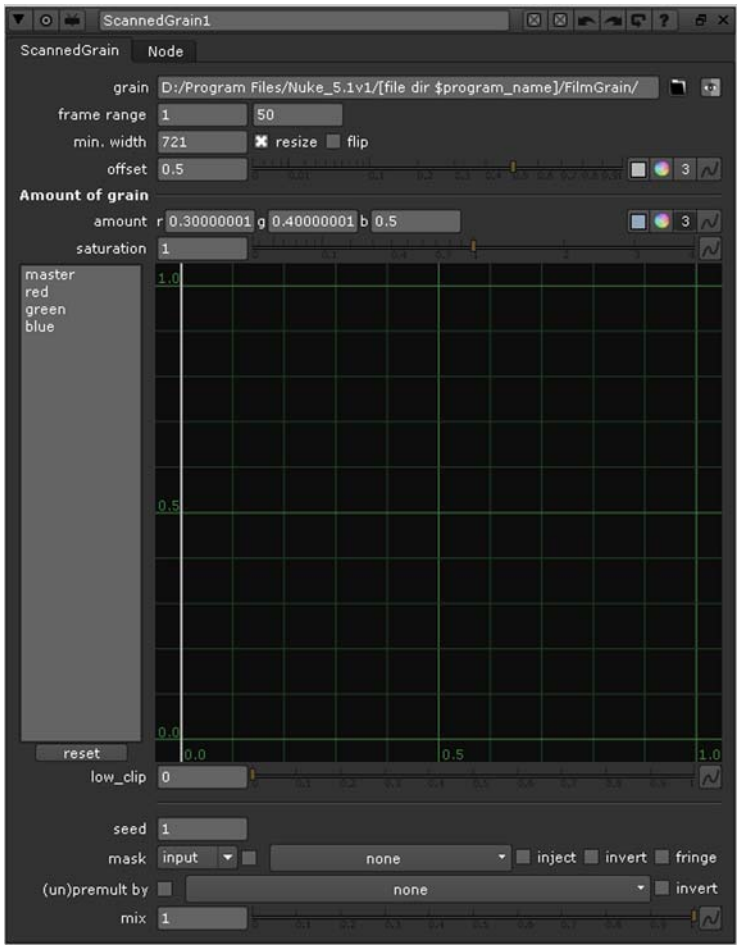


To download film stock sequences:

1. Select **Help** > **Tutorials**.
2. Click on a grain sample to download it. The downloads are in compressed tar format (tgz). The grain samples are .rgb files.

To add scanned grain with the ScannedGrain node:

1. Click **Draw** > **ScannedGrain** to insert a ScannedGrain node at the appropriate place in your script.
2. Connect a Viewer to the output of the ScannedGrain node so you can see the effect of your changes.



3. Click the folder icon of the **grain** field and navigate to the appropriate film stock sequence. Select **Open**.
4. If necessary, check the **resize** box to scale the grain sequence up or down to match your working resolution.
5. In the **min. width** field, define a minimum width (in pixels) that images have to have in order to receive grain.
6. Enter values into the red, green, and blue **amount** fields to increase or decrease on a per-channel basis the density of granules. (This is accomplished, crudely speaking, by boosting or reducing the gain of the grain sequence.)

Now you're ready to fine-tune the blending between the grain and backplate.

To mix the grain and backplate:

1. Drag on the **saturation** slider to increase or decrease the intensity of the grain's hue across all channels.
2. If necessary, you can also use the supplied curve editor to edit the grain sequence's colour curves. In this manner, you can alter gain, gamma, contrast, etc. on a per channel basis. (These curves function in the same manner as those describe in *Using Colour Curves* on page 141).

3. To set a low threshold, based on the input image, below which the grain will not be subtracted, adjust the **low_clip** slider.

Applying Mathematical Operations to Channels

Nuke's **Color** icon in the toolbar houses a number of nodes which are designed to apply common mathematical operations to channels. These operations include clamps, offsets, inversions, multiplications, and expressions.

Clamping Channel Values

To *clamp* a channel's values is to ensure that its blackest blacks and whitest whites will be visible on an intended display device. Nuke's Clamp node lets you assign "legal" values to colours that are either too light or dark for the intended display device.

Figure 4-17: Clamping black and white pixels to "legal" values.

For this effect, you use Nuke's Clamp node.

To clamp channel values:

1. Click **Color** > **Clamp** to insert a Clamp node at the appropriate point in your script.
2. Connect a Viewer to the output of the Clamp node so you can see the effect of your changes.
3. In the Clamp properties panel, use the **channels** field to select the channel you wish to clamp.
4. Drag the **minimum** slider to the legal value. (This has the effect of causing black values to go gray.)
5. Drag the **maximum** slider to the legal value. (This has the effect of causing white values to go gray.)

Offsetting Channel Values

To *offset* a channel's values is to add a fixed value to them, which, in effect lightens the whole channel. You can also add a negative value to a channel, in which case the channel gets darker.

Figure 4-18: Offsetting channel values.

For this effect, you use Nuke's Add node.

To offset channel values:

1. Click **Color** > **Math** > **Add** to insert a Add node at the appropriate point in your script.
2. Connect a Viewer to the output of the Add node so you can see the effect of your changes.
3. In the Add properties panel, use the **channels** field to select the channel you wish to offset.
4. Use the **value** slider to input the value you wish to add to the channel's values.
5. If you are using premultiplied input images, you may want to check **(un)premult by** and select **rgba.alpha** from the pulldown menu. This will simulate doing the addition before the premultiplication was done.

Inverting Channel Values

To *invert* a channel is to subtract its values from one, which causes its blacks to become white and its whites to become black. In the course of building a script, you'll have frequent need to invert mattes in particular.

Figure 4-19: Inverting channel values.

To invert channels you use Nuke's Invert node.

To invert channel values:

1. Click **Color** > **Invert** to insert an Invert node at the appropriate point in your script.

2. Connect a Viewer to the output of the Invert node so you can see the effect of your changes.
3. In the Invert properties panel, use the **channels** field to select the channel you wish to invert.

Multiplying Channel Values

To *multiply* a channel's values is to times them by a given factor, which has the effect of lightening the channel while preserving the blackpoint. (This operation is also knows as gain.)

Figure 4-20: Multiplying channel values.

For this effect, you use Nuke's Multiply node.

To multiply channel values:

1. Click **Color** > **Math** > **Multiply** to insert a Multiply node at the appropriate point in your script.
2. Connect a Viewer to the output of the Multiply node so you can see the effect of your changes.
3. In the Multiply properties panel, use the **channels** field to select the channel whose values you wish to multiply.
4. Use the **value** slider to input the factor by which to you want to times the channel's values.

Applying Expressions to Channel Values

Up till now, the discussion has focused on how to apply simple mathematical formulae—additions, subtractions, multiplications, etc.—to a channel's values. Nuke's Expression node, however allows you to apply complex fomulae to a channel's values. The actual syntax for expressions is rather complex, and thus must be deferred to Chapter 16, *Expressions,* on page 357. For now, you can read about the basics of how to operate the Expression node.

To apply expressions to channel values:

1. Click **Color** > **Math** > **Expression** to insert an Expression node at the appropriate point in your script.
2. Connect a Viewer to the output of the Expression node so you can see the effect of your changes.
3. In the Expression properties panel, use the channel menus and buttons to select the channel to which you wish to apply an expression.
4. Type the actual expression in the **=** field next to the channel.

For example, to assign noise to the red channel, then boost the gain of that result by 20 you would type **(random*r)*20**.

5. If necessary, you can apply different expressions to different sets of channels by repeating the above steps for the other channel menus and buttons.
6. If you need to use a long expression in several fields, you can use the fields on top of the properties panel for assigning the expression temporarily to a variable. Enter your variable on the left side of the = sign, and the expression on the right. You can then use the variable to represent the expression in the = fields next to the channels.

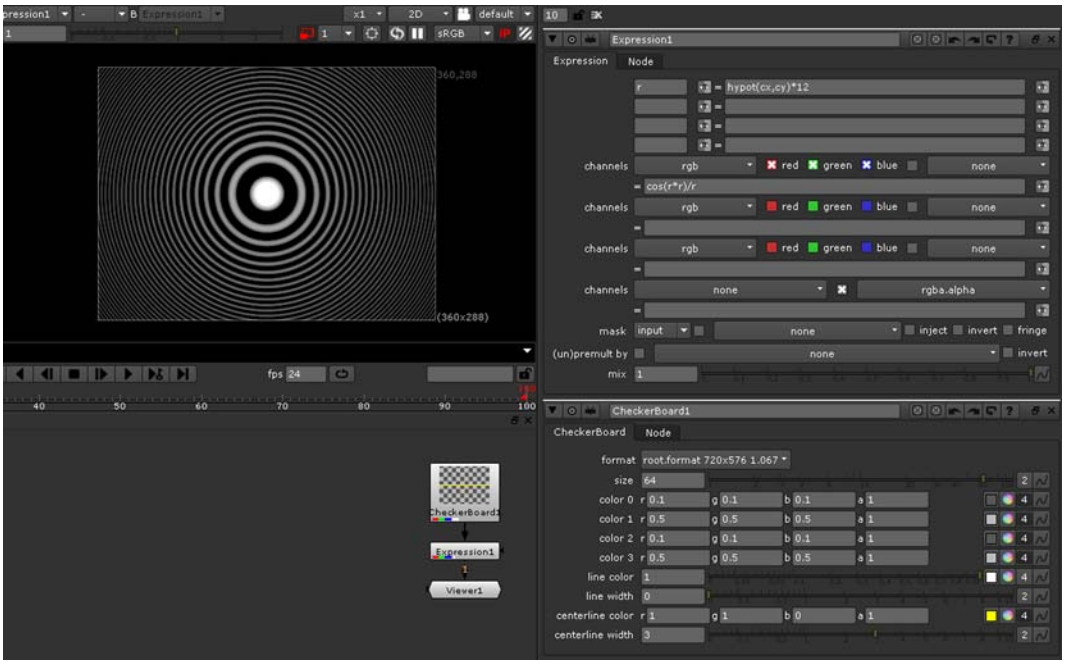


Figure 4-21: A checkerboard modified using an Expression node.

Transforming the Colour Space

Whenever you read a clip into a script, it is automatically converted to Nuke's native colour space, which is 32-bit per channel RGB, a linear format. This conversion takes place even if the clip you read in is in the Kodak Cineon format, which is a logarithmic format.

The reverse of this conversion, called a lin-to-log conversion, also automatically takes place when you write the processed element back out of the script—that is, Nuke automatically converts it back into a Cineon element.

Overriding the Default Cineon Conversion

Nuke uses the Kodak-recommended settings when making Cineon conversions in either direction. It's rare that you would want to override these settings, but if it becomes necessary you can use Nuke's log2lin node.

To override the default Cineon conversions:

1. Double-click on the Read node of the Cineon element whose conversion you wish to override.
2. In the Read properties panel, set the **colorspace** pulldown list to **linear**. This halts the automatic log-to-lin conversion.
3. Click **Color** > **Log2Lin** to insert a log2lin node directly after the Read node.
4. In the log2lin properties panel, set the **operation** pulldown to **log2lin**.
5. Set **black**, **white**, and **gamma** to the appropriate values.
6. Copy the log2lin node and insert it just before the element's Write node.
7. Open up the properties panel of the second log2lin node and set the **operation** pulldown list to **lin2log**. This gives you the reverse of the conversion you created above.
8. Double click on the element's Write node.
9. In the Write properties panel, set the **colorspace** pulldown list to **linear**. This halts the automatic lin-to-log conversion and lets the one you create above have priority.

Making Other Colour Space Conversions

You can also convert elements from Nuke's native colour space to other colour spaces more appropriate to a given process or intended display device. For conversions such as this, use Nuke's Colorspace node, which supports RGB, HSV, YUV, CIE, and CMS formats (and various subformats).

To convert an element in Nuke's native colour space into another colour space:

1. Click **Color** > **Colorspace** to insert a Colorspace node into the appropriate place in your script.
2. In the Colorspace properties panel, set the rightmost pulldown menu in the **out** controls to the appropriate standard.
3. Set the pulldown menu in the middle of the **out** controls to the appropriate standard.
4. Set the leftmost pulldown menu in the **out** controls to the colour space of your choice.
5. If you wish to reverse this conversion later in the script:
   - Copy the Colorspace node and insert it at the appropriate point in your script.
   - Set the **out** controls to **sRGB**, **D55**, and **RGB**.
   - Set the **in** controls to match the values you entered in steps 2, 3, and 4 above.
6. If you wish write out the element in the new colour space:
   - Double click on the element's Write node.
   - In the Write properties panel, set the **colorspace** pulldown list to **linear**. This halts the automatic conversion and lets the one you create above have priority.

Changing the Viewer Colour Space

By default, a script's Viewers display images in Nuke's native colour space. You can, however, set a script's Viewers to display images in non-native colour spaces. This allows you, for example, to see a Cineon element as it appeared prior to the log-to-lin conversion. Changing

the display colour space in no way affects your rendered output. You are applying a display-only lookup table.

For more information on the available LUTs, see *Altering a Script's Lookup Tables (LUTs)* on page 433.

To change the displayed colour space for individual Viewers:

Select the desired colour space from the Viewer's LUT menu on the top right corner.

To change the displayed colour space for all Viewers:

1. Click **Edit** > **Project settings** to display the Settings dialog.
2. Click the **LUT** tab to display the current lookup table settings.
3. Set **viewer** to the desired colour space.
4. Or edit the **lookup** gamma curve to alter the Viewer display. This allows you, in effect, to create a lookup table on the fly.

For example, drag down the **lookup** curves end point to **0.5** to create a lookup table that decreases the brightness of all pixels by half.

5 Transforming Elements

This chapter explains how to perform a range of 2D and 2.5D spatial transformations. You learn how to apply geometric transformations (including translations, rotations, scales, and skews) to elements, and how to add motion blur using the nodes in the Transform menu.

Note that this chapter discusses how to manually apply transformations. Chapter 6, *Tracking and Stabilising* discusses how to use Nuke's tracker to automatically generate and apply transformations.

Transforming in 2D

This section describes how to apply 2D transformations including translations, rotations, scales, and skews to elements using a number of Nuke nodes.

Using the 2D Transformation Overlay

Several of the nodes discussed in this section display a Viewer overlay for executing spatial transformations. This overlay is often a faster alternative to the properties panel. The figure below shows you how to use Nuke 2D transformation overlay.

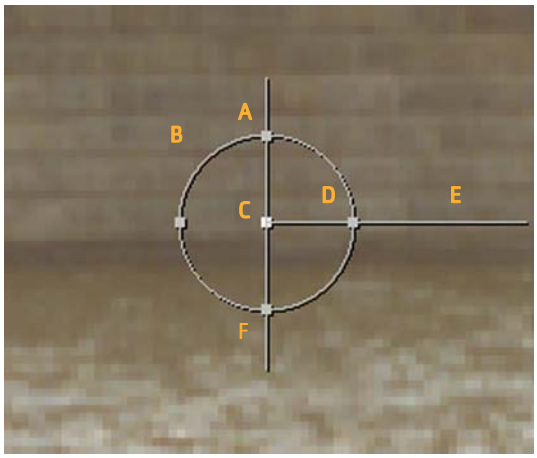


Figure 5-1: 2D Transformation Overlay

A. Drag to skew the frame (see *Skewing Elements* on page 168).

B. Drag to scale the frame uniformly—simultaneously on x and y (see *Scaling Elements* on page 172).

C. Drag to translate the frame (see *Translating Elements* on page 171).

   Shift+drag to constrain the translation to x or y.

   Ctrl/Cmd+drag to reposition the *pivot point* (the point that acts as the centre to transformation operations).

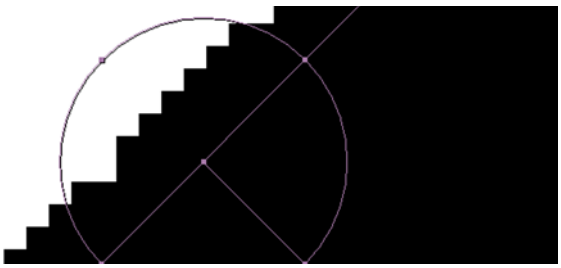
D. Drag to scale the frame on x.

E. Drag to rotate the frame around the pivot point (see *Rotating Elements* on page 172). The transform overlay snaps to typical values. To prevent the snapping, press **Shift** while dragging.

F. Drag to scale the frame on y.

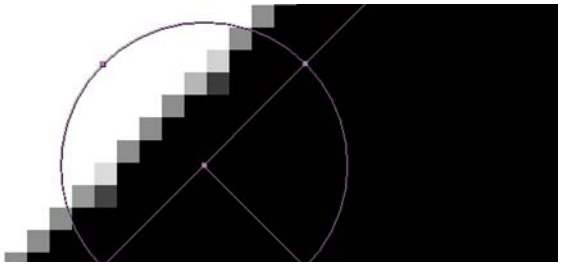
Choosing a Filtering Algorithm

Spatial transformations involve remapping pixels from their original positions to new positions. The question arises as to what values to assign remapped pixels. In the simplest case, they retain their original values, but this can create problems with image quality, particularly in high contrast areas of the frame. For example, the figure below shows a close up a highcontrast feature that has been rotated clockwise by 45 degrees. The remapped pixels have retained their original values, but the result is a highly aliased, or jaggy, edge:



The solution is to apply a more sophisticated *filtering algorithm* to determine the values of remapped pixels—one that takes into account, in some fashion, the values of neighbouring pixels.

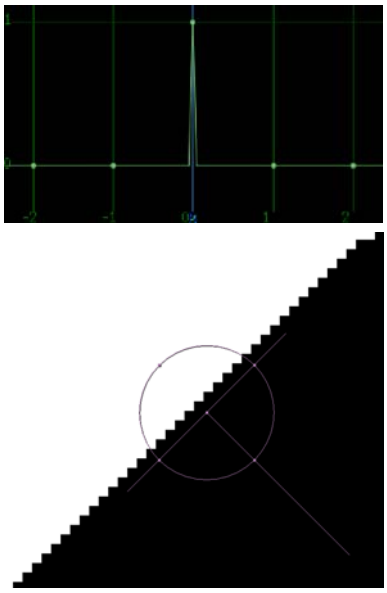
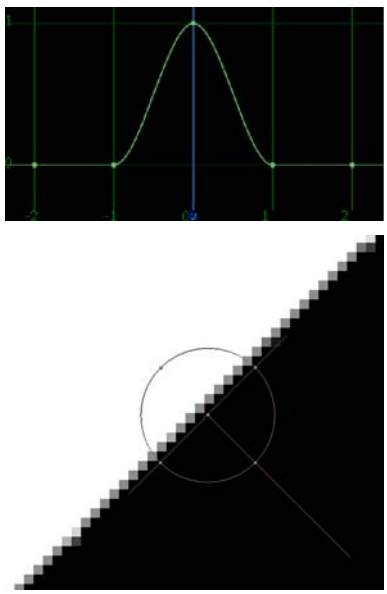
For example, applying Nuke's cubic algorithm to the above rotation, results in a softer, less jagged edge:

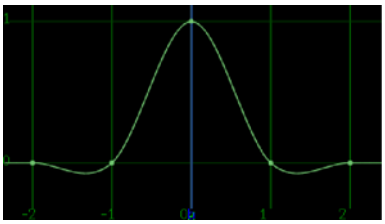
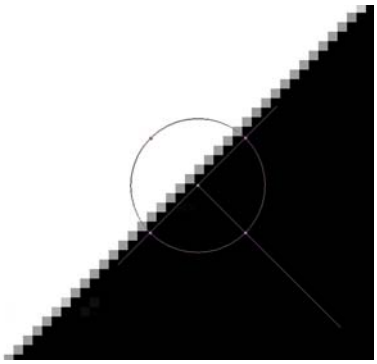
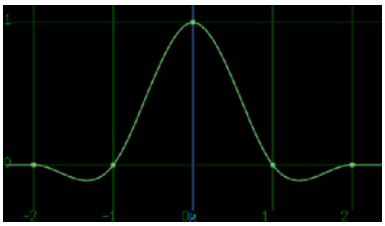
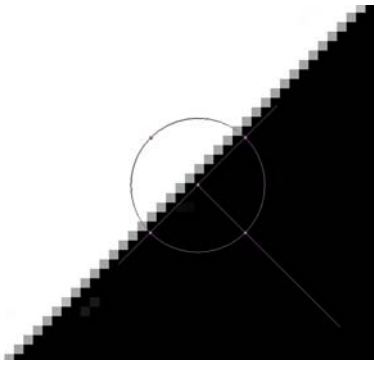


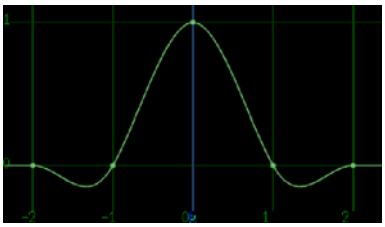
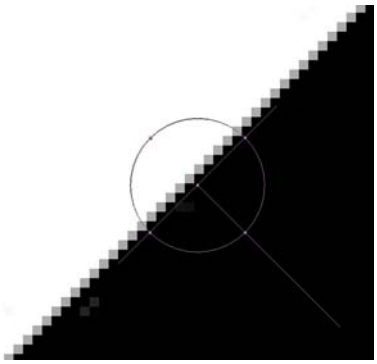
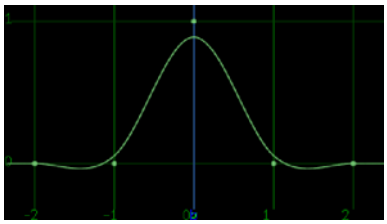
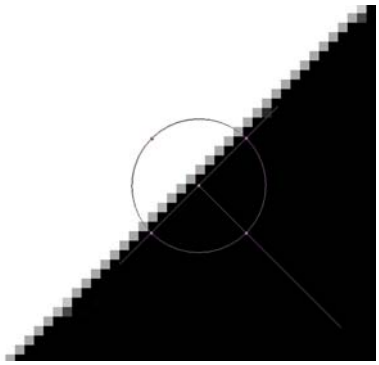
When executing spatial transformations, Nuke lets you choose from the filtering algorithms described in the table below.

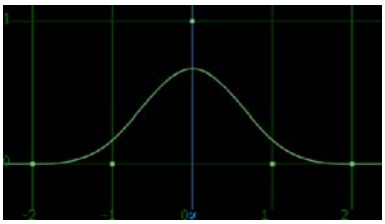
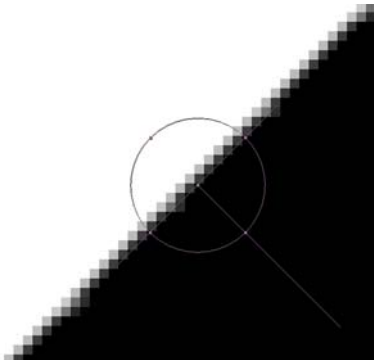
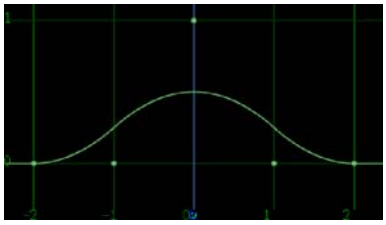
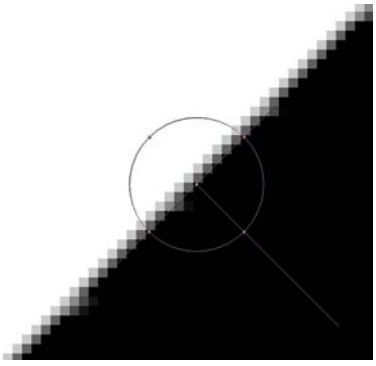
Note that the curves shown in the table plot the manner by which each algorithm samples from neighbouring pixels. The centre of each curve represents the value of the remapped pixel

itself, and the rising and falling portions of each curve represent the amount of sampling that occurs across a five pixel radius.

Filter	Description	Sampling Curve and Output
Impulse	Remapped pixels carry original values.	
Cubic (default)	Remapped pixels receive some smoothing.	

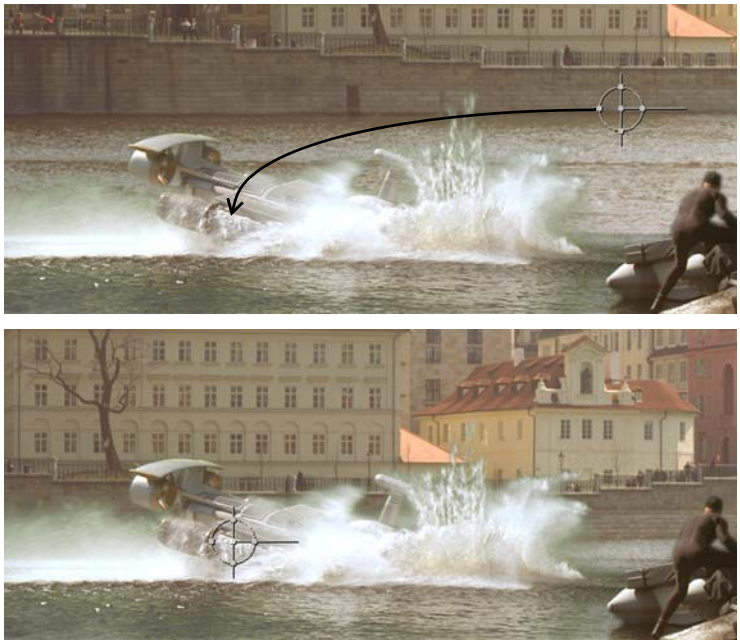
Filter	Description	Sampling Curve and Output
Keys	Remapped pixels receive some smoothing, plus minor sharpening (as shown by the negative -y portions of the curve).	 
Simon	Remapped pixels receive some smoothing, plus medium sharpening (as shown by the negative -y portions of the curve).	 

Filter	Description	Sampling Curve and Output
Rifmen	Remapped pixels receive some smoothing, plus significant sharpening (as shown by the negative -y portions of the curve).	 
Mitchell	Remapped pixels receive some smoothing, plus blurring to hide pixelation.	 

Filter	Description	Sampling Curve and Output
Parzen	Remapped pixels receive the greatest smoothing of all filters.	 
Notch	Remapped pixels receive flat smoothing (which tends to hide *moire* patterns).	 

Translating Elements

To *translate* an element is to slide it on x or y.



You can use the Transform, TransformMasked, or Position nodes to translate elements.

Using the Transform Node

The Transform and TransformMasked nodes let you not only translate elements, but also rotate, scale, and skew them from a single properties panel.

TransformMasked is identical to Transfrom except that it offers controls for assigning a mask to protect certain areas of the frame from translations. For the sake of brevity, this chapter only discusses the use of Transform, but keep in mind you can use TransformMasked any time you need to process a transformation through a mask. Its mask controls work in the same fashion as those described in *Masking Colour Corrections* on page 148.

To translate an element using the Transform node:

1. Click **Transform** > **Transform** to insert a Transform node at appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, increment or decrement the **translate x** and **y** fields to slide the element along either axis.

   Or drag on the centre of the transformation overlay.

Using the Position Node

The Position node gives you just bare-bones parameters for translating an element.

To translate an element using the Position node:

1. Click **Transform** > **Position** to insert a Position node at appropriate place in your script.
2. Connect a Viewer to the output of the Position node so you can see the effect of your changes.
3. In the Position properties panel, increment or decrement the **translate x** and **y** fields to slide the element along either axis.

Rotating Elements

To *rotate* an element is to spin it around the pivot point.

Use the Transform node to rotate elements.

To rotate an element using the Transform node:

1. Click **Transform** > **Transform** to insert a Transform node at appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, choose the appropriate filtering algorithm from the **filter** pulldown list (see *Choosing a Filtering Algorithm* on page 160).
4. Position the pivot point as necessary:
   - Increment or decrement the **center x** and **y** fields to move the axis in either direction.
   - Or press **Ctrl** (**Cmd** on a Mac) while dragging on the centre of the transformation overlay.
5. Increment or decrement the **rotate** field.

   Or drag on the horizontal bar of the transformation overlay.

Scaling Elements

To *scale* an element is to resize it by adding (upsampling) or removing (downsampling) pixels.

Nuke offers several nodes for scaling elements. Transform, whose scaling functions are described below, is designed primarily for scaling up or down the background plate in a composite.

Reformat is designed for writing out elements with specific resolutions and pixel aspect ratios. *Adding Motion Blur* on page 173 describes the use of this node.

To scale an element using the Transform node:

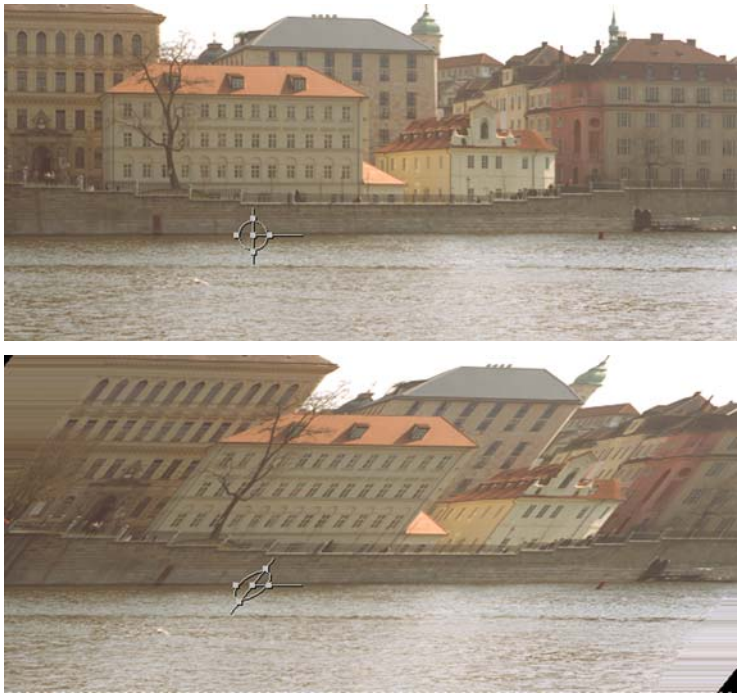
1. Click **Transform** > **Transform** to insert a Transform node at appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, choose the appropriate filtering algorithm from the **filter** pulldown list (see *Choosing a Filtering Algorithm* on page 160).
4. Position the pivot point as necessary:
   - Increment or decrement the **center x** and **y** fields to move the axis in either direction.
   - Or press **Ctrl** (**Cmd** on a Mac) while dragging on the centre of the transformation overlay.
5. To scale the frame uniformly (on both x and y):
   - Increment or decrement the Transform node's **scale** field.
   - Or drag the circle-portion of the of the transformation overlay.
6. To scale the frame asymmetrically (on x or y):

- Click **scale** parameter's channel chooser to reveal the **x** and **y** fields, then increment or decrement each individually.

- Or drag any of the four points on the circle-portion of the transformation overlay. The top and bottom points scale on y; the left and right points, on x.

Skewing Elements

To *skew* an element is to rotate its pixel columns around the pivot point.



Use the Transform node to skew elements.

To skew an element using the Transform node:

1. Click **Transform** > **Transform** to insert a Transform node at appropriate place in your script.
2. Connect a Viewer to the output of the Transform node so you can see the effect of your changes.
3. In the Transform properties panel, choose the appropriate filtering algorithm from the **filter** pulldown list (see *Choosing a Filtering Algorithm* on page 160).
4. Position the pivot point as necessary:
   - Increment or decrement the **center x** and **y** fields to move the axis in either direction.
   - Or **Ctrl**+drag (**Cmd**+drag on a Mac) on the centre of the transformation overlay.
5. Increment or decrement the **skew** field to rotate the pixel columns around the pivot point.

   Or drag the vertical bar of the transformation overlay.

Applying Core Transformations in 2.5D

Nuke's Card3D node lets you apply the same geometric transformations possible with the Transform node, but gives you an additional axis of operation, z.

Just to be clear, the Card3D node's transformations are not truly 3D, but rather what is sometimes called "2.5D"—meaning that you can move an element back on the z axis, but doing so does not convey the sense that it is behind or in front of another element. 2.5D transformations are useful for tasks like "cheating" the perspective of an element or "faking" a camera zoom.

Remember, however, that Nuke doesn't limit you to 2.5 dimensions. If you need true 3D capabilities, you can construct a 3D scene. See Chapter 13, *3D Compositing,* on page 277.

Adding a Card3D Node

To add a Card3D node:

1. Click **Transform** > **Card3D** to insert a Card3D node at appropriate place in your script.
2. Connect a Viewer to the output of the Card3D node so you can see the effect of your changes.

Specifying the Order of Operations

The order by which Nuke executes operations can affect the outcome. The Card3D node lets you choose the order by which Nuke executes scales, rotations, and translations, as well as the order by which it executes rotation on individual axes.

To choose the operation order for scales, rotations, and translations:

In the Card3D properties panel, select an option from the **transform order** pulldown list, which displays all possible combinations (**S** signifies scale, **R**, rotation; and **T**, translation).

To choose the operation order for rotations:

Select an option from the **rotation order** pulldown list, which displays all possible axial combinations.

Choosing a Filtering Algorithm

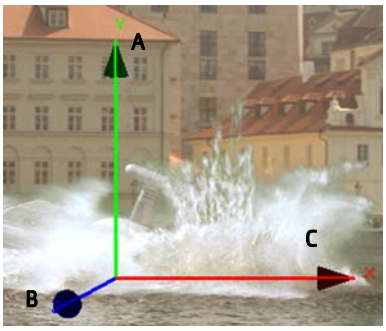
Filtering algorithms let you specify the degree of smoothing and sharpening that remapped pixels receive during transformation. The Card3D node offers the same filter algorithms as the Transform node. See *Choosing a Filtering Algorithm* on page 160 for more information.

To choose a filter algorithm:

Select the desired algorithm from the **filter** pulldown list.

Using the 3D Transformation Handles

You'll note when viewing the output of a Card3D node that it displays an overlay for executing spatial transformations. This overlay is often a faster alternative to the properties panel. The figure below shows you how to use it.



A. Drag to translate the frame on any axis (see *Translating Elements* on page 165).

   Press **Shift** while dragging to constrain the translation to y.

   Press **Ctrl/Cmd** while dragging to rotate the frame on any axis (see *Rotating Elements* on page 166).

   Press **Ctrl/Cmd+Shift** while dragging to constrain the rotation to y.

B. Drag to translate the frame on any axis.

   Press **Shift** while dragging to constrain the translation to z.

   Press **Ctrl/Cmd** while dragging to rotate the frame on any axis.

   Press **Ctrl/Cmd+Shift** while dragging to constrain the rotation to z.

C. Drag to translate the frame on any axis.

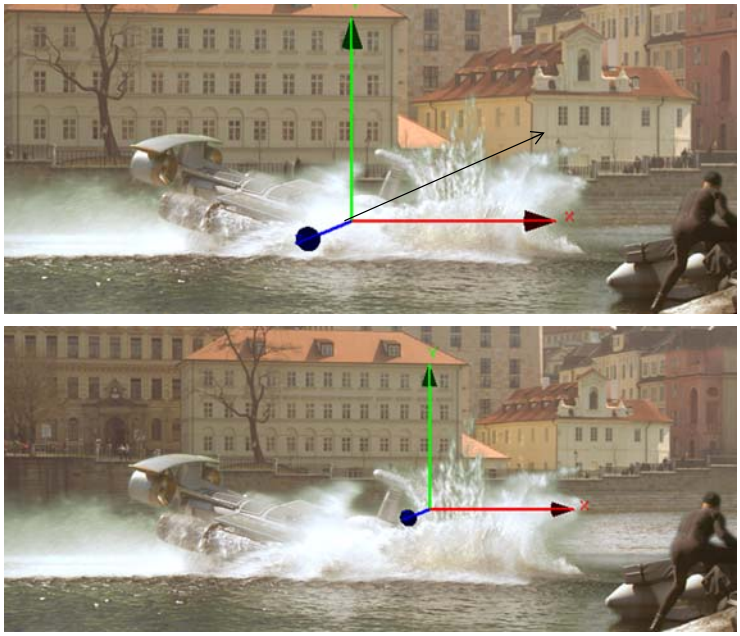
   Press **Shift** while dragging to constrain the translation to x.

   Press **Ctrl/Cmd** while dragging to rotate the frame on any axis.

   Press **Ctrl/Cmd+Shift** while dragging to constrain the rotation to x.

Translating Elements

When using the Card3D node, you can translate elements on z in addition to the other axes.



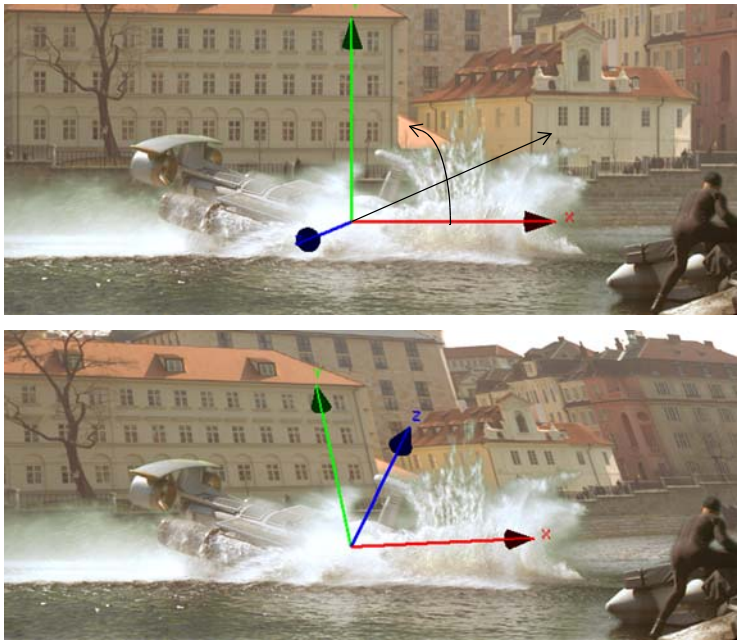
To translate an element using the Card3D node:

In the Card3D properties panel, increment or decrement the **translate x**, **y**, and **z** fields to slide the element along any axis.

Alternatively, you can drag, or press **Shift** while dragging on any axis on transformation overlay.

Rotating Elements

When using the Card3D node, you can rotate elements around the x and y axes, in addition to the z. This is useful for cheating the perspective.



To rotate elements using the Card3D node:

1. Position the pivot point as necessary by incrementing or decrementing the **pivot x**, **y**, and **z** fields to move the axis in any direction.

   Alternatively, you can position the pivot point by pressing **Ctrl/Cmd+Alt** while dragging.
2. Increment or decrement the **rotate x**, **y**, and **z** fields to spin the element around the pivot point.

   Alternatively, you can press **Ctrl/Cmd** while dragging or **Ctrl/Cmd+Shift** while dragging on any axis on the transformation overlay.

Scaling Elements

To scale an element using the Card3D node:

1. Position the pivot point as necessary by incrementing or decrementing the **pivot x**, **y**, and **z** fields to move the axis in any direction.

   Alternatively, you can position the pivot point by pressing **Ctrl/Cmd+Alt** while dragging.
2. To scale the frame simultaneously on x, y, and z, increment or decrement the **uniform scale** field.
3. To scale the frame asymmetrically, increment or decrement the **scale x**, **y**, and **z** fields.

Skewing Elements

Whereas the Transform node lets you rotate pixel columns only around the z axis, Card3D permits you to so around all three axes.

To skew an element using the Card3D node:

1. Position the pivot point as necessary by incrementing or decrementing the **pivot x**, **y**, and **z** fields to move the axis in any direction.

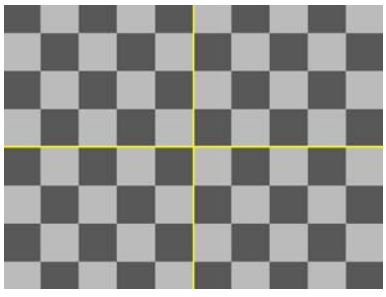
   Alternatively, you can position the pivot point by pressing **Ctrl**/**Cmd**+**Alt** while dragging.
2. Increment or decrement the **skew x**, **y**, and **z** fields to rotate the pixel columns around the corresponding axes.

Adding Motion Blur

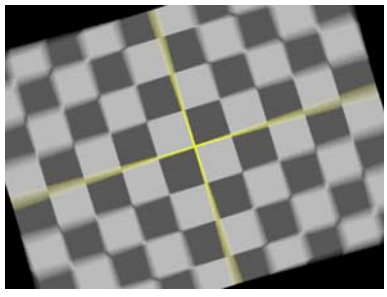
The following nodes under the Transform menu have their own controls for adding motion blur to transformations:

- Transform
- TransformMasked
- Card (3D)
- CornerPin2D
- Reconcile3D
- Tracker
- Stabilize2D.

These controls allow you to create motion blur without adding a separate node for it. The output is similar to a TimeBlur node (see *Applying the TimeBlur Filter* on page 243), but rather than averaging the results of several whole images computed at steps over the shutter period, a number of samples are taken at many random times over the shutter period. This effectively gives many more "steps" and thus a smoother looking result for a smaller total number of computations.



Before rotation and motion blur



After rotation and motion blur

Figure 5-2: Using the Transform node to rotate a checkerboard and add motion blur.

When using several of these nodes in a row, the motion blur is concatenated, and the last transform in the chain defines the motion blur applied.

To add motion blur:

1. Open the transform node's controls.
2. Create a transform and animate it. For instructions on how to do this, see *Animating Parameters* on page 56.
3. In the **motionblur** field, enter the sampling rate. This affects the number of times the input is sampled over the shutter time. The higher the rate, the smoother the result. In many cases, a value of 1.0 is enough. Setting the value to 0 produces no motion blur.
4. In the **shutter** field, enter the number of frames the shutter stays open when motion blurring. For example, a value of 0.5 would correspond to half a frame. Increasing the value produces more blur, and decreasing the value less.
5. From the **shutteroffset** pulldown menu, select when the shutter opens and closes in relation to the current frame value:
   - to centre the shutter around the current frame, select **centred**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter will stay open from frame 29,5 to 30,5.
   - to open the shutter at the current frame, select **start**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter will stay open from frame 30 to 31.
   - to close the shutter at the current frame, select **end**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter will stay open from frame 29 to 30.
   - to open the shutter at the time you specify, select **custom**. In the field next to the pulldown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame, enter a negative value. For example, a value of -0.5 would open the shutter half a frame before the current frame.

6 Tracking and Stabilising

Nuke features a 2D tracker that allows you to extract animation data from the position, size, and rotation of an image. Using expressions, you can apply the data directly to transform and matchmove another element. Or you can invert the values of the data and apply it to the original element—again through expressions—to stabilise the image.

This is the general process for tracking an image:

1. Connect a Tracker node to the image you want to track.
2. Choose the tracking operation you want to perform: stabilise or matchmove.
3. Place tracking anchors over features in the image.
4. Calculate the tracking data.

Before you track, it's important to playback the image several times. This will help you identify the best features for the process, as well as any problems with motion blur or features moving out of frame.

For some images, you may need to filter or colour-correct the image to boost the visibility of features before you attempt to track them. Nuke saves the result as animation data, so you can disable the filter nodes or colour correction after you get a successful track.

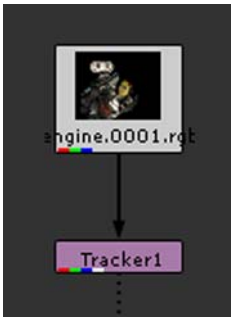
Tracking an Image

The Tracker can analyse the movement of up to four different features in a single image. Nuke generates one animation curve or *track* for each feature.

A single track is usually sufficient to record a feature's horizontal and vertical position across the 2D plane. Two or more tracks are required to extrapolate scaling and rotation.

To track position only (a single feature):

1. Select the node that outputs the image you want to track.
2. Choose **Transform > Tracker** to connect a new Tracker node.

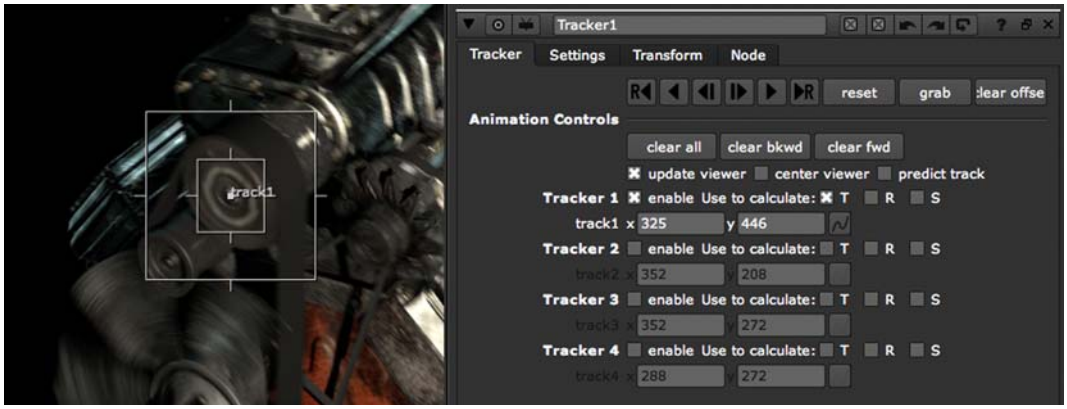


3. Inside the Tracker's properties panel, click the **Transform** tab.
4. From the **transform** list, choose an operation:

   To match another element to the current image, choose **match-move**.

To eliminate movement (i.e., camera shake) from the current image, choose **stabilize**.

5. Click the **Tracker** tab. In the Viewer, you will see the anchor for the first track.



6. Drag the anchor over the feature in the image you want to track.

   When you move the anchor in the Viewer, the **track1, x** and **y** values change to reflect the centre of the anchor.

7. In the properties panel, under **Tracker Controls**, press the track forward button to generate the tracking data from the current frame forward.

To track position, rotation, and scaling (multiple features):

1. Select the node that outputs the image you want to track.
2. Choose **Transform > Tracker** to connect a new Tracker node.
3. Inside the Tracker's properties panel, click the **Transform** tab.
4. Choose and operation from the **transform** list: **match-move** or **stabilize**.
5. Click the **Tracker** tab and check the **enable** box for each of the tracks you want to activate—one for each feature you want to track in the image.

   For example, suppose you want to track the four corners of a billboard so you can matchmove a new image to it—a cornerpin track. you'll need four tracks activated.

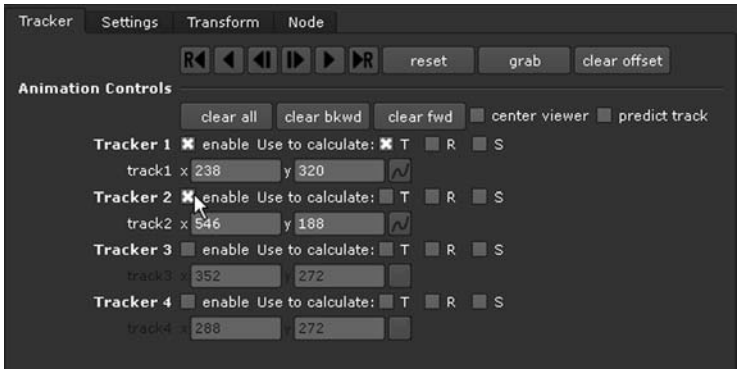
6. Drag the anchors over the features in the image you want to track.
7. Press the track forward button in the Tracker properties panel to generate the animation data for all enabled tracks.

Activating Track Anchors

You can select up to four track anchors. The number you choose depends on which transformational components you wish to track and the degree of accuracy you require.

To activate the tracks:

1. Click the **Tracker** tab in the properties panel.
2. Check each of the boxes for the tracks you want to **enable**.



Note

After you calculate a track, you can uncheck its enable box to lock it. This protects the tracked points from being recalculated or repositioned.

Positioning Track Anchors

A *pattern* and *search area* accompany each track anchor. The pattern area encompasses the grid of pixels that the system attempts to follow across multiple frames. This pattern should be as distinct as possible from the surrounding frame, and remain visible throughout the majority of the sequence. For example, you might choose as a pattern a high-contrast window corner which stays in frame throughout an entire shot.

The search area defines the portion of the frame in which the system looks for the pattern.

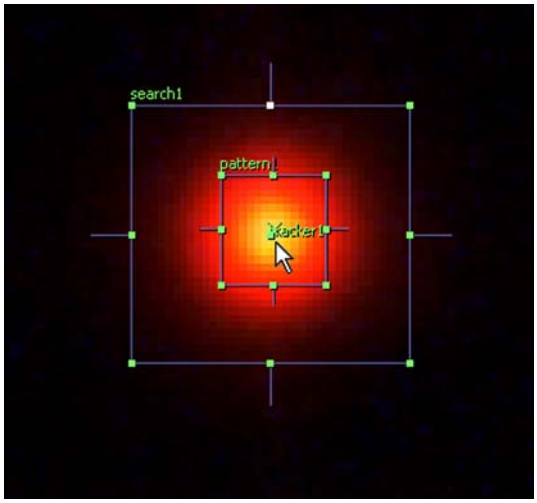


Figure 6-1: The search area contains the space where the tracker will search for the pattern. The pattern area contains the pixels that the tracker will attempt to "lock onto" for the track.

Positioning track anchors involves moving and sizing the boundaries of both the search and pattern areas. Start by moving both boundaries over the pattern to be tracked, then fine tune the position and size of each. In the end, the search area must be larger than the pattern area.

To move both the search and pattern boundaries:

1. Drag on the frame to select both boundaries with the marquee.
2. Click on the border of either boundary, then drag both over the pattern to be tracked (stop when the pattern boundary overlay's **x** sits directly on top the feature).

To adjust the position of either the search or pattern boundaries.

1. Click to the line-portion of either boundary to select it.
2. Drag to reposition the boundary.

   Or, if you're repositioning the pattern boundary, increment or decrement the track's **x** and **y** fields.

To adjust the size of the search or pattern boundaries.

1. Click on any point on either boundary.
2. Drag to reposition the associated side.

Calculating the Track

Once you've properly placed the track anchors and sized the search and pattern areas, you're ready to calculate the track(s). You calculate tracks by using the buttons under **Tracker controls** in the Tracker properties panel. You can track the sequence in either direction. Tracking backwards often allows you to get accurate tracks when a feature exhibits a lot of scaling—when the distance between the feature and the camera changes.

To toggle the tracking overlay:

- If necessary, turn on the Tracker overlay in the Viewer. Click the right mouse button and choose **Overlay on/off**, or just press **O,** over the Viewer.

  Pressing **O** three times toggles between the three overlay states: **overlay off**, **overlay on**, and **overlay on, no animation path**.

To calculate tracks:

1. In the Tracker properties panel, check the **enable** box for each track you wish to calculate.
2. In the Tracker properties panel, click either the frame forward or backward buttons to move to the previous or next frame. Move through a few frames in this manner to ensure that all enabled track anchors are "sticking" to their patterns.

3. If a particular track anchor doesn't stick, experiment with a different position.
4. Once all track anchors stick, click the Tracker's track forward or track backward buttons to analyse the whole sequence.

When calculating multiple tracks simultaneously, you may find that some tracks stick with accuracy to the pattern, while others require resetting and reanalysis. When you're happy with a given track, uncheck its **enable** box. This protects it from recalculation, and let's you experiment with better placement for the wayward tracks.

If you need to start over with a given track anchor, you can reset the size of its search and pattern boxes and wipe its existing tracking data.

To reset the size of an anchor's search and pattern boxes:

1. Check the **enable** box for only the track anchor whose size you wish to reset.
2. Click the **reset** button. The track anchor's pattern and search areas are recentred to their default sizes.

To clear a track's animation data:

1. Check the **enable** box for only the track anchor whose track you wish to remove.
2. Under **Animation Controls**, click the **clear all** button. The selected track is removed—that is, all it's transformational data is wiped.

   To only clear animation forward or backward of the current frame, click **clear fwd** or **clear bkwd**.

Retracking Part of a Track

A tracking pattern may become unusable when it moves out of frame, is hidden by another image feature, or because of motion blur. When this happens, you can retrack the unusable part of the track with new search and pattern areas while keeping the track data consistent. The end result is a continuous track calculated from multiple patterns.

To retrack part of a track with a new search area:

1. Check the **enable** box for only the track that requires retracking.
2. Cue the Viewer to the last frame where the existing tracking is usable.
3. **Ctrl**+drag (**Command**+drag on Mac OS X) the track anchor to reposition the search and pattern areas without affecting the position of the track point. The offset allows Nuke to continue the track with the assumption that the offset feature remains at the same relative distance to the original feature.
4. Click the Tracker's track forward (or backward button, if you are tracking backwards) to continue calculating the track using the new pattern. Because the track point has been offset from the new search area, the new track values continue smoothly from the existing ones.

Editing Tracks

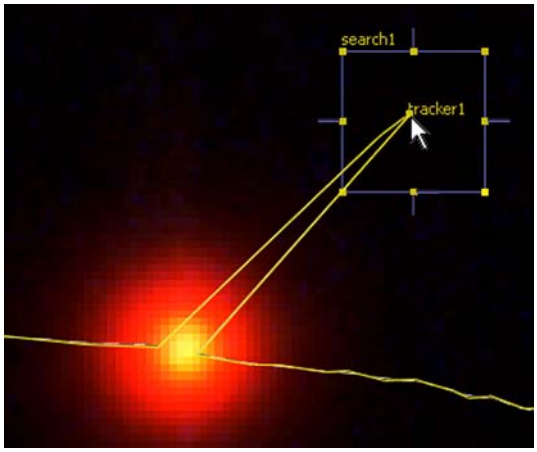
You can edit tracks via their Viewer overlays or their underlying animation curves.

Manipulating the Track Overlays

The Tracker plots existing tracks as Viewer overlays. These overlays offer an intuitive means of editing a track. If for example, you have a track in which the Tracker loses sight of its pattern for one or two frames, you can use the overlay to manually reposition the wayward track points. (This is often a faster approach than going into the associated animation curves.)

To move track points with the overlay:

1. Cue the Viewer to the frame corresponding to the track point you wish to move. The search and pattern boxes move over the point.
2. Drag the track point to the desired location. The search and pattern boxes follow.



Manipulating Track Curves and Smoothing Tracks

A track is essentially just an animated transformation matrix. Thus each track has animation curves which you can edit in order to refine a track. You can also smooth tracks using the Tracker controls.

Moving Track Points with Curves

To move track points with curves:

1. In the Tracker properties panel, click the animation button next to the track you wish to edit, then select **Curve Editor**. The Animation editor displays the x and y curves for the track (these plot the position of each track point over time).

2. Select the points on these curves which you wish to manipulate. (Click to select individual points; drag to select multiple points with the marquee; or press **Crtl+A** to select all points.).
3. Drag the points to adjust their values. As you do so, the tracker overlay on the Viewer changes shape to reflect the new positions of the track points.

Smoothing Tracks

Once applied to an element, some tracks may exhibit too much jitter, which is caused by the Tracker too precisely following the pattern. You can use the Tracker controls or apply smoothing filters to a track's curves in order to remove such jitter.

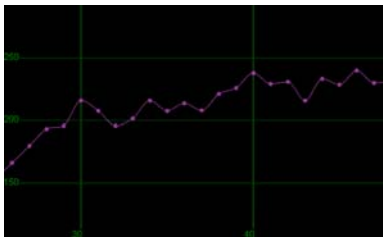
To smooth tracks:

1. In the Tracker controls, go to the **Transform** tab.
2. In the **smooth** fields, enter the number of frames you want to average together to smooth the transformation. You can smooth the translate (**T**), rotate (**R**), and scale (**S**) separately.

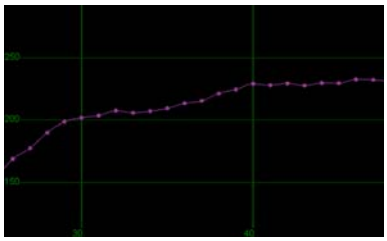
OR

1. In the Tracker properties panel, click the animation button next to the track you wish to edit, then select **Curve Editor**. The Animation editor displays the x and y curves for the track (these plot the position of each track point over time).

2. Select the points on these curves which require smoothing. (Click to select individual points; drag to select multiple points with the marquee; or press **Crtl+A** to select all points.).
3. Right-click on the editor and select **Edit > Filter** to apply the smoothing filter. This sets new values on each point based on the average values of their neighbouring points.
4. Enter the number of times to apply the smoothing filter in the dialog that appears. Click **OK**.
5. Reapply the smoothing filter as many times as is necessary.



Before smoothing



After smoothing

Figure 6-2: A track curve before and after smoothing.

Tracking and Multiview Projects

If you need to use tracking data in a multiview or stereoscopic project, you may want to apply your edits to one view only (for example, the left view but not the right), or create a track in one view and have it automatically generated for the other, in the correct position.

Splitting Views Off

Splitting a view off allows you to edit the tracking data in that view only, without affecting any other views that exist in your project settings.

To split a view off:

1. Display the view you want to split off in the Viewer.
2. In the Tracker controls, display the **Tracker** tab. Usually, it is better to split off controls on this tab rather than the **Transform** tab. The controls on the Transform tab will compute differently per view as long as the tracks on the **Tracker** tab are different per view.
3. Click the view button next to the track you want to edit and select **Split off [view name]**. For example, to edit tracking data in a view called left, select **Split off left**. Any changes you now make to the track in question are only applied to the view you chose to split off and are displaying in the Viewer.

Correlating One View from Another

You can use a Tracker to track something in one view, and have the track's x and y position automatically generated for the other view.

To correlate one view from the other:

1. Track a feature in one view.
2. In the Viewer, display the view you want to generate the corresponding track for.

3. In the Tracker controls, click the view button next to the track, and select **Correlate [view name] from [view name] using disparity**. For example, if you created a track for the left view and want to have the corresponding track generated for the right view, select **Correlate right from left using disparity**. This generates the corresponding track for the view you are displaying.

Tip

If you have got The Foundry's Ocula plug-ins installed, you can also do the correlation using Ocula (select **Correlate [view name] from [view name] with Ocula**). This way, extra refinements are done when generating the track, and the results may be more accurate.

For more information on working with multiview projects, see Chapter 14: *Working with Stereoscopic Projects* on page 329.

Applying Tracking Data

You apply tracking data to the input image or other Nuke nodes using either the Tracker node's controls or linking expressions.

Applying Tracking Data Using Tracker Controls

The simplest way to apply tracking data to the input image or other nodes is to use the controls of the Tracker node itself. Here, we look at using these controls to stabilise or matchmove footage. If you need to apply a cornerpin track to another node, you need to do it via linking expressions.

Stabilising Elements

The Tracker node's controls let you remove motion, such as unwanted camera shake, from the node's input clip.

To stabilise the input footage:

1. Create the track you want to use for stabilising the footage. A single track is usually enough to stabilise a feature's horizontal and vertical motion across the 2D plane. Two tracks can be used to do the same but also remove rotation in the image.
2. In the Tracker properties panel, go to the **Settings** tab. From the warp type pulldown menu, select the transformations that you want Nuke to take into account when stabilising the image, for example **Translate/Rotate/Scale**.
3. Go to the **Transform** tab. Under **transform**, select **stabilize**.

Nuke stabilises the footage, locking its elements to the same position within the composite.

Matchmoving Elements

You can use the Tracker node's controls to apply the tracked motion to another image, that is, to matchmove an image.

To matchmove footage:

1. Use a Tracker node to create the track you want to apply to an image.

2. Copy the Tracker node and paste it after the footage you want to matchmove.
3. In the second Tracker node's controls, go to the **Transform** tab.
4. From the transform pulldown menu, choose **match-move**.

Nuke applies the tracked movement to the footage you want to matchmove.

Applying Tracking Data Via Linking Expressions

Nuke's CornerPin2D and Stabilize2D nodes are specifically designed to receive tracking data via linking expressions, but you can apply tracking data in this manner to virtually any Nuke node. For example, you might animate a Bezier shape with tracking data by entering linking expressions into the Bezier node's transformation parameters. You can also apply tracking data to individual Bezier points.

This section explains the basic procedure for applying tracking data to any node via linking expressions, then discusses how to apply such data to the CornerPin2D and Stabilize2D nodes in particular.

Creating Linking Expressions

The Tracker node's Tracker panel displays data related to the position of each track anchor over time (**tracks' x** and **y** fields). These are the data which you most typically apply to other nodes.

To drag and drop tracking data:

1. Display both the tracker parameters (the source parameters, in this case) and the parameters to which you wish to apply the tracking data (the destination parameters—for example, a Bezier node's **translate** parameter).
2. **Ctrl**+drag (**Cmd**+drag on a Mac) from the source parameters animation button to the destination parameters animation button.

When you release, the destination parameters will turn blue, indicating an expression has been applied. In this case, the drag and drop action has created a linking expression resembling the following example:

```
Tracker1.tracker1.x
```

Tip

You can also apply tracking (or other transform) data to individual Bezier, SplineWarp, or GridWarp points (this is sometimes called per vertex tracking). To do so, **Ctrl**/**Cmd**+drag and drop the track's animation button on a Bezier, SplineWarp or GridWarp point in the Viewer.

You can add other components to this linking expression as necessary. For example, you might add a spatial offset to the linking expression by subtracting out the initial frame's tracking values, in which case the final expression would resemble the following:

```
Tracker1.tracker1.x-Tracker1.tracker1.x(1)
```

See Chapter 16: *Expressions* on page 357 for more information. Once you enter the linking expression, the destination parameter turns blue.

Using the CornerPin Node

The CornerPin2D node is designed to map the four corners of an image sequence to positions derived from tracking data. In practice, this node lets you replace any four-cornered feature with another image sequence. For example, suppose you needed to replace the monitor image in the fast-panning shot shown below.

Figure 6-3: Fast-panning shot requires four corner tracking.

You would first use the Tracker to calculate four separate tracks, one for each corner of the feature.

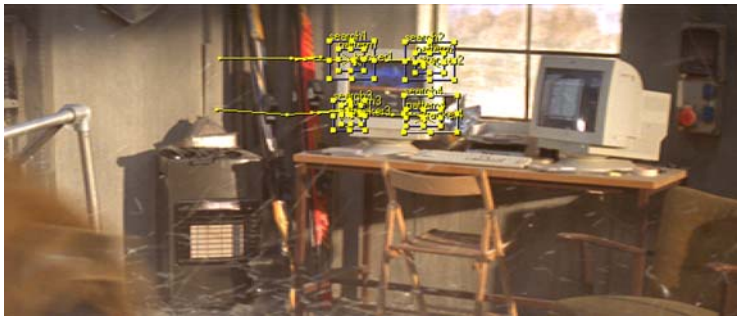


Figure 6-4: Generating the four tracks.

Next, you would attach a CornerPin2D node to the image sequence you want to use as the replacement for the feature, and apply to it the tracking data. This would remap the image sequence's corners to the correct positions over time

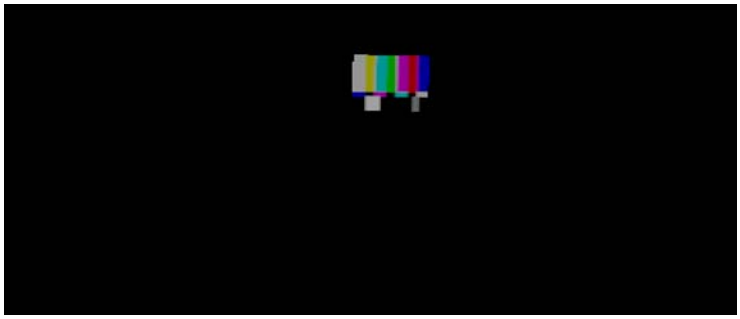


Figure 6-5: Applying the tracked corner data to the replacement image.

The final step would be to layer this result over the original element.

Figure 6-6: The composited image.

The steps below summarize the use of the CornerPin2D node.

To use the CornerPin2D node:

1. Use the Tracker node to generate four tracks, one per corner, on the feature requiring replacement.
2. Click **Transform > CornerPin** to add a CornerPin2D node to the script.
3. Attach the CornerPin2D node to the image sequence that will replace the feature tracked above.
4. In the CornerPin2D properties panel, add linking expressions (see *Creating Linking Expressions* on page 183) to the positional data for the four tracks generated above.

   When linking a particular track to a particular corner, keep in mind that **to1** refers to the bottom left corner of the image sequence; **to2**, to the bottom right corner; **to3**, to the top right corner, and **to4**, to the top right corner.
5. If necessary, choose a different filtering algorithm from the **filter** pulldown menu. (See *Choosing a Filtering Algorithm* on page 160).
6. When filtering with Key, Simon, or Rifmen filters, you may see a haloing effect caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.
7. In most cases, you will keep **black outside** checked. This renders as black pixels outside the image boundary, making it easier to layer the element over another. (If you uncheck this parameter, the outside area is filled with the outermost pixels of the image sequence.)

Using the Stabilize2D Node

The Stabilize2D node is designed to remove unwanted camera movement, rotation, and/or scaling from an image sequence. The node requires data from only a single track if you only need to stabilise movement; it requires data from two tracks if you need to stabilise for rotation and/or scaling.

The basic procedure for using Stabilize2D is to first use the Tracker node to generate the required tracks, then follow the Tracker node with a Stabilize2D node. To this node, you apply the tracking data in inverse form, thus negating the unwanted transformations.

To use the Stabilize2D node:

1. Use the Tracker node to generate the appropriate number of tracks on the element requiring stabilisation. Remember, you'll need at least two tracks if you need to stabilise for more than just movement. (You can, of course, generate more tracks and average the results for better accuracy.)
2. Select the Tracker node used above, then click **Transform > Stabilize** to add a Stabilize2D node to the Tracker node.
3. From the **type** pulldown menu in the Stabilize2D properties panel, select:
   - **1 Point** to stabilise only for movement.
   - **2 Point** to stabilise for rotation and/or scaling.
4. Check all transformation types which you wish to cancel out.
5. In the both the **track1 x** and **y** fields, type **1-** (to invert the data), followed by a linking expression to the relevant tracking data in the Tracker node used above. (You can use either the positional data for Tracker 1, or some multiple track average from the Outputs panel.)

   Your entries should resemble the following examples: **1-Tracker1.tracker1.x** and **1-Tracker1.tracker1.y**.
6. Repeat the above for the **track2 x** and **y** fields.
7. As you apply the tracking data, the current frame displayed in the Viewer is likely to move out of view. This is because the node applies the inverted tracking data to the bottom left corner of the image sequence. Enter values in the **offset XY x** and **y** fields to restore the image to the centre of frame. (You may have to animate these values over time to keep the image centred.)
8. If necessary, choose a different filtering algorithm from the **filter** pulldown. (See *Choosing a Filtering Algorithm* on page 160).
9. When filtering with Key, Simon, or Rifmen filters, you may see a haloing effect caused by pixel sharpening these filters employ. If necessary, check **clamp** to correct this problem.
10. In most cases, you will keep **black outside** checked. This renders as black pixels outside the image boundary. (If you uncheck this parameter, the outside area is filled with the outermost pixels of the image area.)

7 Primatte

This section explains how to use the blue/green screen keyer, Primatte, in Nuke.

Accessing Primatte from Nuke

Start up Nuke, create a Primatte node and connect a foreground and a background image to it. Add a Nuke Viewer node so you can see the result.

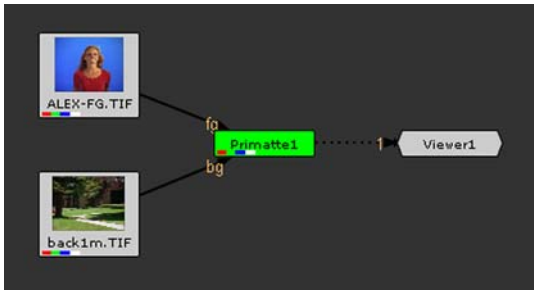


Figure 7-1: Nuke flowgraph.

When you select the Primatte node, you should be presented with the Primatte properties panel as shown below.

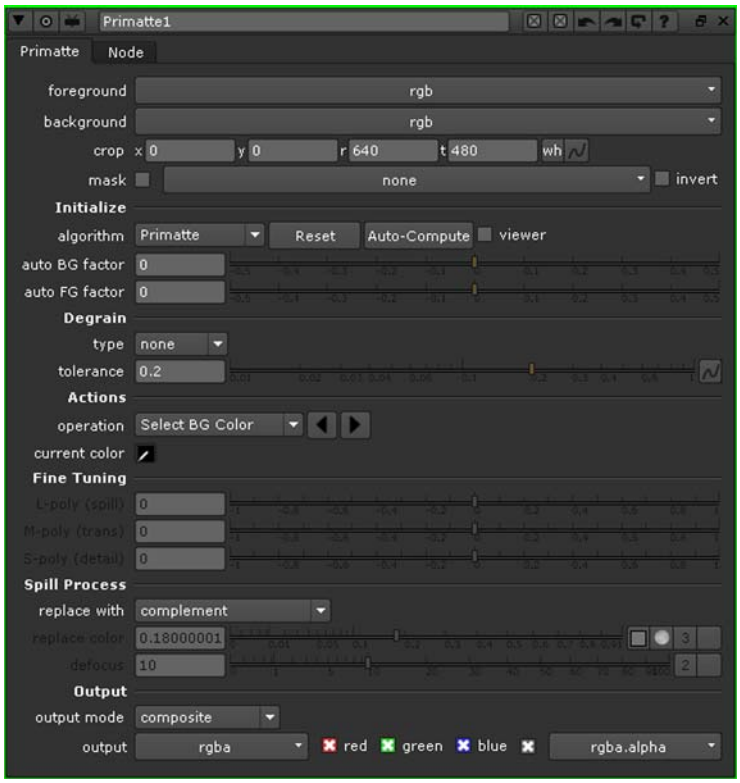


Figure 7-2: Primatte properties panel.

Primatte Basic Operation Tutorial

This describes the operation of the Primatte node in Nuke. A more detailed explanation of how the Primatte algorithm actually works can be found under *The Primatte Algorithm* on page 210.

Auto-Compute

This version of Primatte has a new feature that may eliminate the first three steps of using earlier versions of Primatte. It is called the **Auto-Compute** button and may make your keying operation much easier. You can click on this button as a first step and it may automatically sense the backing screen colour, eliminate it and even get rid of some of the foreground and background noise that would normally be cleaned up in Step 2 (**Clean BG Noise**) and Step 3 (**Clean FG Noise**) of the Primatte operation. If you get good results then jump ahead to the spill removal tools.

The **Auto-Compute** button has two sliders that modify its behaviour; the **auto BG factor** and the **auto FG factor** sliders. These may be moved to get better results with the **Auto-Compute** button. This is useful when doing a set of clips that have similar backgrounds and lighting. Once the sliders are configured for a particular lighting set-up, all the clips will key quickly using just the **Auto-Compute** button.

If you don't get the results you wanted from **Auto-Compute**, please continue from this point on to get the basic Primatte operation procedures.

The basic functionality for the Primatte interface is centred around the **Actions** or **operation** pulldown menu and the Viewer window.

Figure 7-3: Primatte operation menu.

There are four main steps to using the Primatte and **Select BG Colour** is the first step.

Select BG Colour

Ensure that the **Select BG Colour** action is selected (it should be at this time as it is the default **Action** mode when you start Primatte).

Position the cursor in the bluescreen area (or whatever background colour you are using), usually somewhere near the foreground object. Hold the **Ctrl/Cmd** key down and sample the targeted background colour. Release the mouse button and Primatte will start the compositing process. If the foreground shot was done under ideal shooting conditions, Primatte will have done 90-95% of the keying in this one step and your image might look like this.

Figure 7-4: Basic key.

Note

Primatte will work equally well with any colour backing screen. It does not have to be a specific shade of green or blue.

Tip

If you dragged the cursor in the blue area, Primatte averages the multi-pixel sample to get a single colour to adjust to. Sometimes Primatte works best when only a single pixel is sampled instead of a range of pixels. The colour selected at this point in the Primatte operation is critical to the operation of the node from this point forward. Should you have difficulties further along in the tutorial after selecting a range of blue shades, try the Select BG Colour operation again with a single dark blue pixel or single light blue pixel. You can also switch to the alpha channel view and click around in the bluescreen area and see the different results you get when the initial sample is made in different areas.

Tip

If you would rather make a rectangular selection and not use the default 'snail trail' sampling method, you can do a **Crtl+Shift+**drag sample.

Tip

If the foreground image has a shadow in it that you want to keep it in the composite, do not select any of the dark blue pixels in the shadow and the shadow will come along with the rest of the foreground image.

The second and third steps in using Primatte require viewing the **Matte or Alpha** view in the **Viewer** window. Press the 'A' key on the keyboard to change to the **Alpha** view. The image displayed will change to a black and white 'matte' view of the image that looks like this.

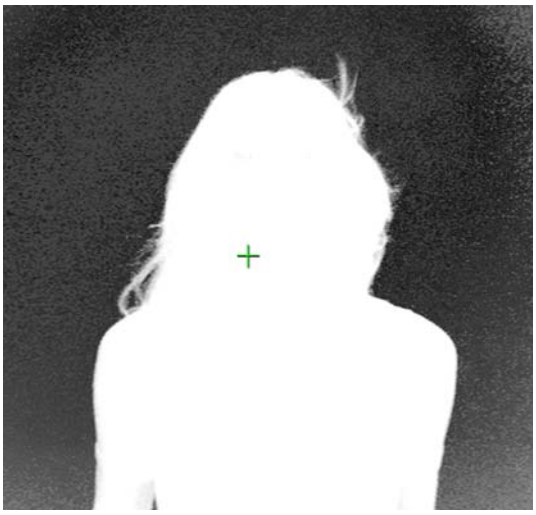


Figure 7-5: Matte.

Clean BG Noise

Change the **Actions Operation** from **Select BG Colour** to **Clean BG Noise**. If there are any white regions in the dark, 'bluescreen area', it is 'noise' (or shades of blue that did not get picked up on the first sample) and should be removed. Sample through these whitish noise regions and when you let up on the pen or mouse button, Primatte will process the data and eliminate the noise. Repeat this procedure as often as necessary to clear all the noise from the background

areas. Sometimes increasing the brightness of your monitor or the screen gamma allows you to see noise that would otherwise be invisible.

Before *After*

Figure 7-6: Before and after background noise removal.

Note

You do not need to remove every single white pixel to get good results. Most pixels displayed as a dark colour close to black in a key image will become transparent and virtually allow the background to be the final output in that area. Consequently, there is no need to eliminate all noise in the blue-screen portions of the image. In particular, if an attempt is made to meticulously remove noise around the foreground object, a smooth composite image is often difficult to generate.

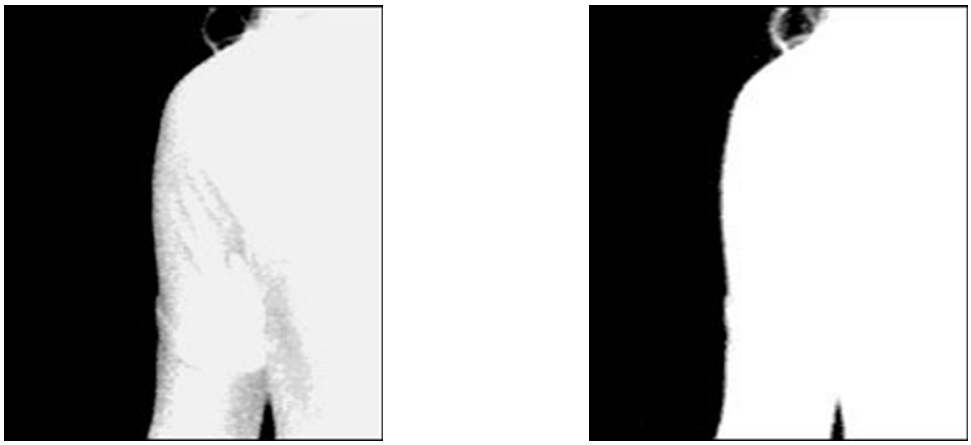
Tip

When clearing noise from around loose, flying hair or any background/foreground transitional area, be careful not to select any of areas near the edge of the hair. Leave a little noise around the hair as this can be cleaned up later using the **Fine Tuning Sliders** tool.

Clean FG Noise

If there are dark regions in the middle of the mostly white foreground object, that is, if the key is not 100% in some portion of the targeted foreground, choose **Clean FG Noise** from the

Actions operation pop-up menu. Use the same techniques as for **Clean BG Noise**, but this time sample the dark pixels in the foreground area until that area is as white as possible.



Before *After*

Figure 7-7: Before and after foreground noise removal.

These were the steps necessary to create a clean 'matte' or 'key' view of the image. With this key, the foreground can be composited onto any background image. However, if there is `spill' on the foreground object from light that was reflected off the background, a final operation is necessary to remove that background spill get a more natural looking composite.

For the fourth step in the Primatte operation, return the **RGB** view to the monitor window by clicking again on the **`A'** keyboard key. This will turn off the alpha channel viewing mode and the Viewer window will again display the **RGB** view with the background image (if you connected one to the Primatte node).
The sample image below has gone through the first three steps and has examples of spill. Notice the blue fringe to her hair and a blue tint on her right cheek, arm and chest.

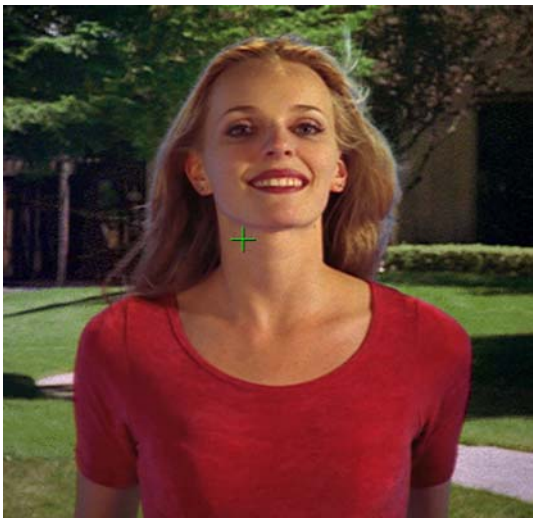


Figure 7-8: Blue spill visible.

Spill Removal – Method #1

There are three ways in Primatte to remove the spill colour. The quickest method is to select the **Spill Sponge** button from the **Actions operation** area and then sample the spill areas away. By just positioning the cursor over a bluish pixel and sampling it, the blue will disappear from the selected colour region and be replaced by a more natural colour. Additional spill removal should be done using the **Fine Tuning** tools or by using the **Spill(-)** feature. Both are explained further on in this manual.

Note

All spill removal/replacement operations in Primatte can be modified using the **Spill Process 'replacement with'** tools. Spill can be replaced with either the **complement** of the background colour, a **solid colour** selected by the user or by colours brought from a **defocused background.** Depending on the spill conditions, one of these options should provide the results you are looking for. See the information in *Spill Replacement Options* on page 196 for more details.

Note

Primatte's spill removal tools work on 'colour regions'. In the image above, samples should be made on the light flesh tones, the dark flesh tones, the light blonde hair, the dark blonde hair and the red blouse colour regions. One sample in each colour region will remove spill from all similar colours in the foreground image.

If the spilled colour was not been totally removed using the **Spill Sponge** or the result of the **Spill Sponge** resulted in artifacts or false colouring, a fine-tuning operation **Spill(-)** tool should be used instead for a more subtle and sophisticated removal of the spilled background colour. This is discussed in *Spill (-)* on page 207.

Spill Removal – Method #2

1. Select the **Fine Tuning Sliders Actions operation**. This will activate these tools.

2. Using the zoom and pan capabilities of the Nuke application, zoom into an area that has some blue edges or spill.
3. Using the cursor, sample a colour region that has some spill in it. When you let up on the pen or mouse button, Primatte will register the colour selected (or an average of multiple pixels) in the **current colour** area. For most images, the **L-poly (spill)** slider is all that is required to remove any remaining blue spill. The more to the right the slider moves, the more spill colour will be removed from the sampled pixels. The more to the left the slider moves, the more the selected pixels will move toward the colour in the original foreground image.

   Note: When using the **L-poly (spill)** slider, spill colour replacement will be replaced based on the setting of the **Spill Process 'replacement with'** settings. For more information on

these tools, see the section of this manual on **Spill Replacement** in **Chapter 6. Primatte Tools and Buttons**.

Tip

It is better to make several small adjustments to the blue spill areas than a single major one.

4. You can use the other two sliders in the same way for different key adjustments. The **S-poly (detail)** slider controls the matte softness for the colour which is closest to the background colour. For example, you can recover lost rarefied smoke in the foreground by selecting the **Fine Tuning Sliders** action, sampling the area of the image where the smoke just starts to disappear and moving the **S-poly (detail)** slider to the left. The **M-poly (trans)** slider controls the matte softness for the colour which is closest to the foreground colour. For example, if you have thick and opaque smoke in the foreground, you can make it semi-transparent by moving the **Transparency** slider to the right after selecting the pixels in the **Fine Tuning Sliders** mode.

Tip

If the foreground image changed colour dramatically during the fine tuning process, you can recover the original colour by selecting an area of the off-colour foreground image and moving the L-poly (spill) slider slightly to the left. This may introduce spill back into that colour region. Again, use the Fine Tuning Sliders option to suppress the spill, but make smaller adjustments this time.

Spill Removal – Method #3

1. This method uses a more recent Primatte tool that is covered in detail in *Repeatable Sampling Tools* below.

Note

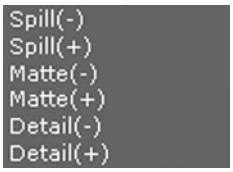
If these final 'spill suppression' operations have changed the final compositing results, you may have to return to earlier operations to clean up the matte. If the composite view looks good, it is a good idea to go back and take a final look at the alpha channel view. Sometimes in the Primatte operation, a 100% foreground area (all white) will become slightly transparent (gray). You can clean those transparent areas up by using the Matte Sponge tool. After selecting the Matte Sponge tool, just click on the transparent pixels and they will become 100% foreground. All of the spill-suppression information will remain intact. Alternatively, you can go to the alpha channel view and then using the Fine Tuning Sliders option, select those transparent areas and move the Transparency slider slightly to the left. This will move that colour region from 0-99% foreground with spill suppression to 100% foreground with spill suppression and should solve the problem. The Matte(+) tool will also work to solve this problem.

Repeatable Sampling Tools

Most of the Primatte operations are done using a 'mouse or pen sampling' operation. The only exceptions are the **Fine Tuning Sliders** Actions mode and its sliders. The **Fine Tuning Sliders** action mode gives a continuous valuator for fine-tuning but some of the sliders are not often used because results are often unpredictable or not subtle enough.

Another weak point in previous versions of Primatte is the lack of functionality to attenuate and thicken the existing matte density. This version of Primatte offers a more intuitive, easy-to-use and powerful user interface called **Repeatable Sampling**.

In addition to the conventional Primatte operation modes previously mentioned, six other tools are added:



The Spill Sampling Tools

Using the **Spill(+)** and **Spill(-)** modes, you can gradually remove or recover the spill intensity on the foreground object by sampling the referenced colour region repeatedly. The conventional **Spill Sponge** tool removes the spill component in a single action at one level and did not allow sampling the same pixel a second time. Even though just a small amount of spill needed to be removed, the spill sponge removed a preset amount without allowing any finer adjustment.

Figure 7-9: The effect of Spill (+/-) repeatable sampling.

Using the zoom and pan capabilities of the Nuke application, zoom into an area that has some blue edges and click on a pixel with some spill on it. Repeated clicking will incrementally remove the spill. Continue this operation until the desired result is achieved.

The Matte Sampling Tools

The **Matte(+)** and **Matte(-)** modes are used to thicken or attenuate the matte information. If you want a thinner shadow on a foreground object, you can use the **Matte(-)** mode as many times as you like to make it more transparent. On the other hand, you can use the **Matte(+)** mode to make the matte thicker in that colour region.

Figure 7-10: Effect of Matte (+/-) Repeatable Sampling.

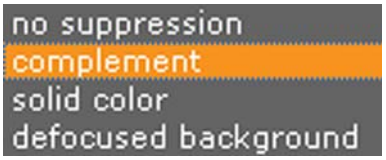
The Detail Sampling Tools

The **Detail(+)** and **Detail(-)** modes are a refined version of **Clean BG Noise** and **Restore Detail** (discussed later in this document). For example, when you see some dilute noise in the backing area but don't want to remove it completely because it affects some fine detail in a different area, try using **Detail(-)**. It will attenuate the noise gradually as multiple samples are made on the pixel. You should stop the sampling when important fine details start to disappear.

Figure 7-11: Effect of Detail (+/-) Repeatable Sampling.

Spill Replacement Options

The proper processing of spill on foreground objects is one of the many useful features of Primatte. You can move between these four modes to see how they affect the image clip you are working with. The four methods are as follows:



1. No Suppression (no suppression)
2. Complemental Spill Replacement (complement)
3. Solid Colour Spill Replacement (solid colour)
4. Defocus Spill Replacement (defocused background)

No Suppression (no suppression):

In this mode, no suppression is applied.

Complemental Replacement Mode (complement):

This is the default spill replacement mode. This mode will maintain fine foreground detail and deliver the best quality results. If foreground spill is not a major problem, this mode is the one that should be used.

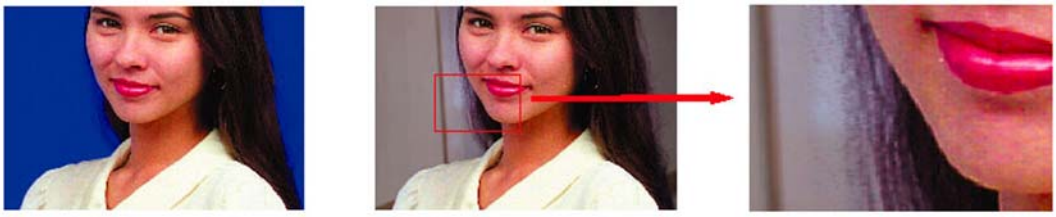


Figure 7-12: Complemental Replacement mode maintains fine detail.

The **Complemental Replacement** mode is sensitive to foreground spill. If the spill intensity on the foreground image is rather significant, this mode may often introduce serious noise in the resultant composite.

Figure 7-13: Serious noise in the composite.

Solid Colour Replacement Mode (solid colour):

In the **Solid Colour Replacement** mode, the spill component will be replaced by a 'user defined' palette colour. While the **Complemental Replacement** mode uses only the backing colour complement to remove small amounts of spill in the original foreground, the **Solid Colour Replacement** mode tries to assuage the noise using the 'user defined' palette colour. Changing the palette colour for the solid replacement, the user can apply good spill replacement that matches the composite background. Its strength is that it works fine with even serious blue spill conditions.

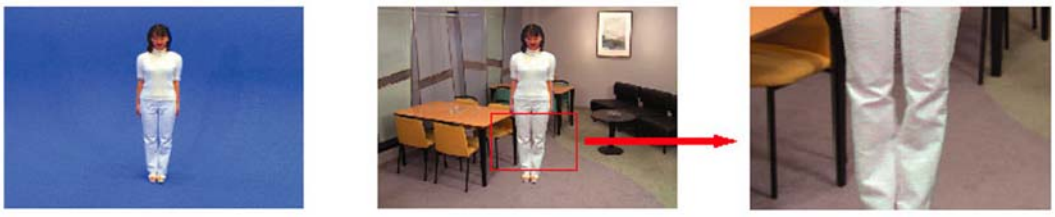


Figure 7-14: Smooth spill processing with solid colour replacement.

On the negative side, when using the **Solid Colour Replacement** mode, fine detail on the foreground edge tends to be lost. The single palette colour sometimes cannot make a good colour tone if the background image has some high contrast colour areas.

Defocus Spill Replacement (defocused background):

The **Defocus Replacement** mode uses a defocused copy of the background image to determine the spill replacement colours instead of a solid palette colour or just the complement colour. This mode can result in good colour tone on the foreground object even with a high contrast background. As in the example below, spill can even be removed from frosted glass using this feature and still retain the translucency.

On the negative side, the **Defocus Replacement** mode sometimes results in the fine edge detail of the foreground objects getting lost. Another problem could occur if the user wanted to later change the size of the foreground image against the background. Since the background/foreground alignment would change, the applied colour tone from the defocused image might not match the new alignment.

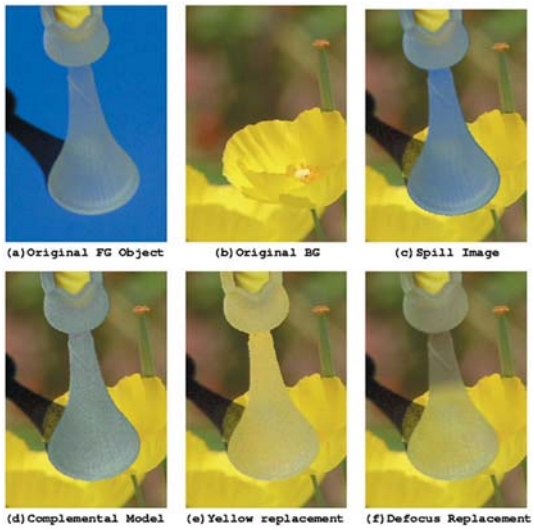


Figure 7-15: Blue suppression of a frosted glass object.

Primatte Tools and Buttons

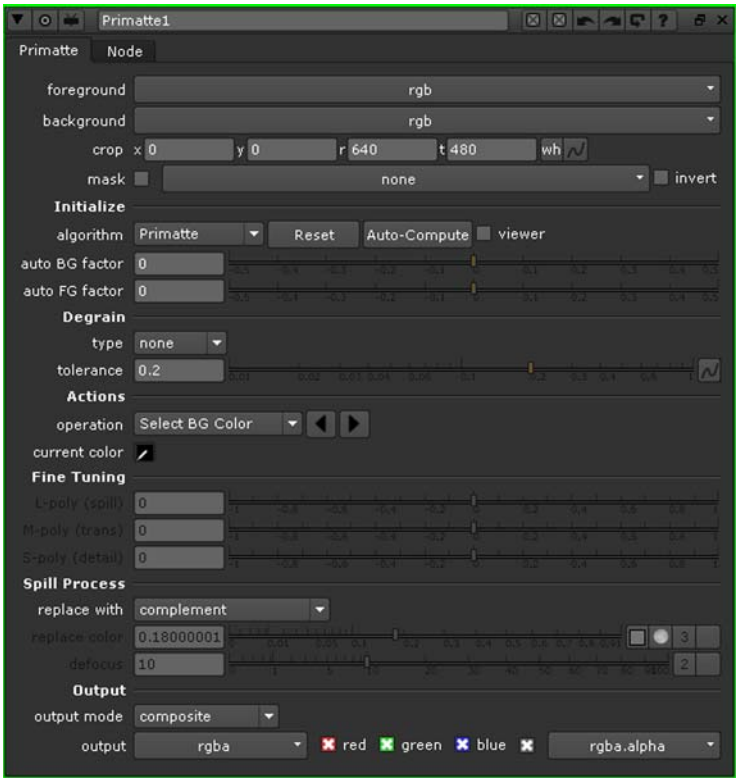
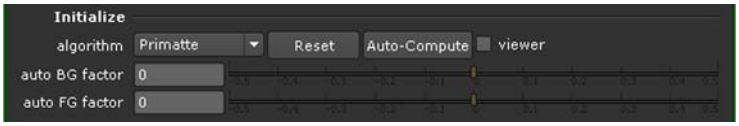


Figure 7-16: Primatte node controls.

Initialize Section



Primatte Algorithm Menu

Primatte Algorithm

The **Primatte** algorithm mode delivers the best results and supports both the **Solid Colour** and the **Complement Colour** spill suppression methods. It is the algorithm that uses three multi-faceted polyhedrons (as described further down in the this document) to separate the 3D RGB colorspace. It is also the default algorithm mode and, because it is computationally intensive, it may take the longest to render.

Primatte RT+ algorithm

Primatte RT+ is in between the above two options. It uses a six planar surface colour separation algorithm (as described further down in the this document) and will deliver results in

between the other two in both quality and performance. Other disadvantages of the **Primatte RT+** algorithm is that it does not work well with less saturated backing screen colours and it does not support the **Complement Colour** spill suppression method.

Primatte RT Algorithm

Primatte RT is the simplest algorithm and therefore, the fastest. It uses only a single planar surface to separate the 3D RGB colorspace (as described further down in the this document) and, as a result, does not have the ability to separate out the foreground from the backing screen as carefully as the above Primatte algorithm. Other disadvantages of the **Primatte RT** algorithm is that it does not work well with less saturated backing screen colours and it does not support the **Complement Colour** spill suppression method.

Reset

Resets all of the Primatte key control data back to a blue or greenscreen.

Auto-Compute

The **Auto-Compute** button can be used as the first step in the Primatte operation. It's purpose is to try and do the first three steps of the Primatte operation for you. It will try to automatically detect the backing screen colour, remove it and do some clean-up on the foreground and background noise. If the clip was shot with an evenly lit, well saturated backing screen, the **Auto- Compute** button will leave you with an image that may only need some spill removal to complete your keying operation.

auto FG Factor

The **auto FG Factor** slider can be used to modify how the Auto-Compute algorithm deals with foreground noise. Change the position of this slider and you can see the results of the **Auto-Compute** operation change.

auto BG Factor

The **auto BG Factor** slider can be used to modify how the Auto-Compute algorithm deals with background noise. Change the position of this slider and you can see the results of the **Auto-Compute** operation change.

3D Viewer

This selector opens a window in the Viewer that displays a graphical representation of the Primatte algorithms and allows the user to see what is happening as the various Primatte tools are used. It is a passive feature that has no adjustment capabilities, it may prove useful in evaluating an image as you operate on it.

When you select it, you are presented with a window that may look similar to one of these images (depending on which Primatte algorithm you have selected).

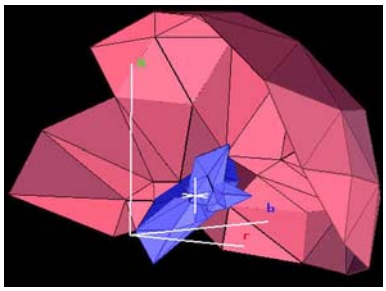
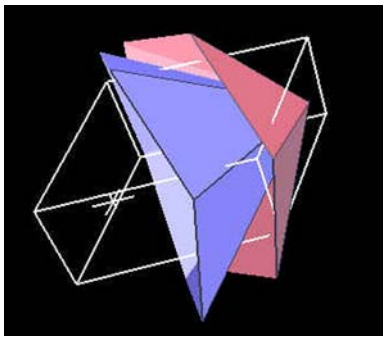
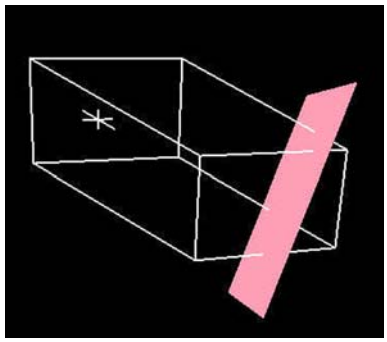


Figure 7-17: Primatte algorithm.



Primatte RT+ algorithm



Primatte RT algorithm

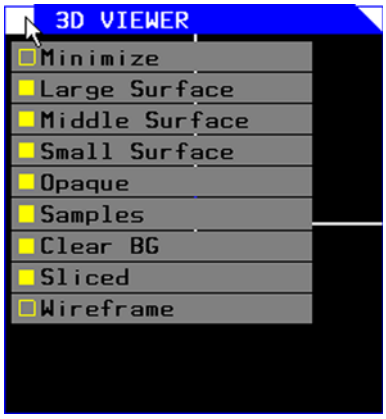
Figure 7-18: Primatte RT+ and Primatte RT algorithms.

The different algorithms are described in more detail in a later section of this manual. Here is a description of the tools and features of the 3D Viewer:

3D Viewer Tools

At the top of the 3D Viewer window are three areas that can be clicked on:

1. Clicking and dragging on the blue centre area allows the user to move the window around on the screen.
2. Clicking and dragging on the triangular white region in the upper right corner allows the user to scale the 3D Viewer window.
3. Clicking on the square white region in the upper left of the window displays a pop-up menu that looks like this:



Note

A 'selected feature has a solid yellow square next to it. An 'unselected' feature has a hollow yellow square next to it.

Minimize

This feature, when selected, makes the 3D Viewer window disappear. Only the blue title bar at the top of the window remains.

Large Surface

This feature, when selected, displays the large Primatte polyhedron in the Viewer window.

Middle Surface

This feature, when selected, displays the medium Primatte polyhedron in the Viewer window.

Small Surface

This feature, when selected, displays the small Primatte polyhedron in the Viewer window.

Opaque

This feature, when selected, makes the selected polyhedrons opaque. De-selecting it makes them semi-transparent.

Samples

This feature, when selected, allows the user to sample colour regions on the image window using the 3D Sample **Actions** mode and see where those regions are in relation to the polyhedron and the algorithm. The colours will be displayed as a spray of pixels in the colour selected. This button only allows the user to see or hide the sampled colours.

Note

The 3D Sample Actions mode must be selected in the Actions area for this feature to operate.

Clear BG

This feature changes the background colour of the 3D Viewer window from black (when unselected) to transparent (when selected).

Sliced

This feature, when selected, slices open the large and medium polyhedrons so that the inner polygons can be seen. When unselected, the largest polyhedron selected becomes a completely closed polyhedron and you might not be able see the inner polyhedrons (unless the **Opaque** feature is deselected).

Wireframe

This feature, when selected, changes the polyhedrons from shaded-surface objects to wireframe objects.

Degrain Section

Degrain Tools

The **Degrain** tools are used when a foreground image is highly compromised by film grain. As a result of the grain, when backing screen noise is completely removed, the edges of the foreground object often become harsh and jagged leading to a poor key. These tools were created to, hopefully, help when a compositing artist is faced with a grainy image.

Degrain type

The **Degrain type** selector gives the user a range of grain removal from 'none' to 'large'. If the foreground image has a large amount of film grain induced pixel noise, you may lose a good edge to the foreground object when trying to clean all the grain noise with the **Clean BG Noise Actions** mode. These tools allow the user to clean up the grain noise without affecting the quality of the key. A short tutorial explaining when and how to use these tools is at the end of this section.

none

When **none** is selected, the user gets the colour of the exact pixel sampled. This is the default mode.

small

When **small** is selected, the user gets the average colour of a small region of the area around the sampled pixel. This should be used when the grain is very dense.

medium

When **medium** is selected, the user gets the average colour of a medium-sized region of the area around the sampled pixel. This should be used when the grain is less dense.

large

When **large** is selected, the user gets the average colour of a larger region of the area around the sampled pixel. This should be used when the grain is very loose.

Tolerance Slider

Adjusting the **tolerance** slider this should increase the effect of the **Clean BG Noise** tool without changing the edge of the foreground object.

Degrain Tools Tutorial

If you have a noisy image as in the example below...

...you will find that the matte is also noisy:

Currently you can use the **Clean BG Noise** operation to remove the noisy pixels, but this can also modify the edge of the foreground object in a negative manner.

Using the **Degrain Tools** in the following way may help you clean up the image and still get a good edge on the matte:

1. Use the **Clean BG Noise** operation just a small amount to remove some of the white noise in the **Alpha** channel view but do use it so much that you affect the edge of the foreground object.
2. Then select the **Grain Size** tool and select **small** as a first step to reduce the grain:

With the degrain **tolerance** slider set at **0**, move it around some. This should increase the affect of the **Clean BG Noise** tool without changing the edge of the foreground object.

Sometimes this may not be enough to totally remove the grain so by adjusting the degrain **tolerance** slider, you can tell the Primatte algorithm what brightness of pixels you think represents grain. You should try not to use too high of a value otherwise it will affect the overall matte. For an example of an 'over adjusted' image see below.

The Primatte degrain algorithm uses a 'Defocused Foreground' image to compute the noise.

Note

The **small**, **medium** and **large** settings for the degrain tools all produce defocused foregrounds that have larger or smaller blurs respectively.

Note

It is important to make sure that the crop settings are correctly applied otherwise when the defocus image is generated, if there is 'garbage' on the edges of the images, then that garbage will be blurred into the defocus foreground.

As a review:

1. Select the **Select BG Colour Actions** mode and click on a backing screen colour.
2. Select the **Clean BG Noise Actions** mode and use it sparingly so that it has minimum affect to the edge of the foreground object.
3. If there is still grain in the backing screen area, then use the degrain **type** functionality starting at the **small** setting to reduce the grain
4. If the grain is still present, then try increasing the **tolerance** slider a little - not too much.
5. If grain is still a problem, then try changing the **type** to **medium** or **large** and also changing the grain tolerance until the desired effect is achieved.

Note

The grain functionality does not always remove grain perfectly but is sometimes useful to minimize its effects.

Actions Section

Actions operation Tools/Modes

Select Background Colour

When this operational mode is selected, the Primatte operation will be initially computed by having the user sample the target background colour within the image window. For keying operations, this is the first step and should be followed by the steps described immediately below.

Clean Background Noise

When this operational mode is selected, the user samples pixels on the image window known to be 100% background. White noisy areas in the 100% background region will become black. This is usually the second step in using Primatte.

Clean Foreground Noise

When this operational mode is selected, the user samples pixels on the image window known to be 100% foreground. The colour of the sampled pixels will be registered by Primatte to be the same colour as in the original foreground image. This will make dark gray areas in the 100% foreground region become white. This is usually the third step in using Primatte.

Spill Sponge

When this operational mode is selected, the background colour component in the sampled pixels (or spill) within the image window is keyed out and removed for the colour region selected. This operation can only be used once on a particular colour and the amount of spill suppression applied is not adjustable. It is the fastest way to remove spill from a composite image. For more accurate spill suppression, a **Fine Tuning** or **Spill (+)** operation should follow or be used instead. This can usually be the fourth (and final) step in using Primatte unless additional adjustments are necessary.

Matte Sponge

When this operational mode is selected, the sampled colour within the image window becomes 100% foreground. However, if the sampled colour is already keyed out and removed, it leaves the current 'suppressed' colour. It only affects the key or matte information. This tool is usually used to quickly remove stray transparent pixels that have appeared during the chromakeying procedure. It is a quick and easy way to make final adjustments to a composite.

Restore Detail

With this mode selected, the completely transparent background region sampled in the image window becomes translucent. This operation is useful for restoring lost hair details, thin wisps of smoke and the like. It shrinks the small polyhedron slightly.

Make Foreground Transparent

When this mode is selected, the opaque foreground colour region sampled in the image window becomes slightly translucent. This operation is useful for the subtle tuning of foreground objects which are otherwise 100 percent covered with smoke or clouds. It can only be used one time on a particular colour. For a more flexible way to thin out a colour region and be able to make multiple samples, you should use the **Matte (-)** tool. It expands the medium polyhedron slightly.

Spill (+)

When this operational mode is selected, colour spill will be returned to the sampled pixel colour (and all colours like it) in the amount of one Primatte increment. This tool can be used to move the sampled colour more in the direction of the colour in the original foreground image. It can be used to nullify a **Spill (-)** step. This tool dents the Primatte large polyhedron in the colour region sampled.

Spill (-)

When this operational mode is selected, colour spill will be removed from the sampled pixel colour (and all colours like it) in the amount of one Primatte increment. If spill colour remains, another click using this operational mode tool will remove more of the colour spill. Continue using this tool until all colour spill has been removed from the sampled colour region. This tool expands the Primatte large polyhedron in the colour region sampled.

Matte (+)

When this operational mode is selected, the matte will be made more opaque for the sampled pixel colour (and all colours like it) in the amount of one Primatte increment. If the matte is still too translucent or thin, another click using this operational mode tool will make the sampled colour region even more opaque. This can be used to thicken smoke or make a shadow darker to match shadows in the background imagery. It can only make these adjustments to the density of the colour region on the original foreground image. It can be used to nullify a **Matte (-)** step. This tool dents the Primatte medium polyhedron in the colour region sampled.

Matte (-)

When this operational mode is selected, the matte will be made more translucent for the sampled pixel colour (and all colours like it) in the amount of one Primatte increment. If the matte

is still too opaque, another click using this operational mode tool will make the sampled colour region even more translucent. This can be used to thin out smoke or make a shadow thinner to match shadows in the background imagery. This tool expands the Primatte medium polyhedron in the colour region sampled.

Detail (+)

When this operational mode is selected, foreground detail will become less visible for the sampled pixel colour (and all colours like it) in the amount of one Primatte increment. If there is still too much detail, another click using this operational mode tool will make more of it disappear. This can be used to remove smoke or wisps of hair from the composite. Sample where is visible and it will disappear. This is for moving colour regions into the 100% background region. It can be used to nullify a **Detail (-)** step. This tool expands the Primatte small polyhedron in the colour region sampled.

Detail (-)

When this operational mode is selected, foreground detail will become more visible for the sampled pixel colour (and all colours like it) in the amount of one Primatte increment. If detail is still missing, another click using this operational mode tool will make detail more visible. This can be used to restore lost smoke or wisps of hair. Sample where the smoke or hair just disappears and it will return to visibility. This is for restoring colour regions that were moved into the 100% background region. It may start to bring in background noise if shooting conditions were not ideal on the foreground image. This tool dents the Primatte small polyhedron in the colour region sampled.

Current Colour Chip

This shows the current colour selected (or registered) by the **Fine Tuning** operational mode.

Fine Tuning Section

Fine Tuning Sliders

When this operational mode is selected, the colour of the sampled pixel within the Viewer window is registered as a reference colour for fine tuning. It is displayed in the **Current Colour Chip** in the **Actions** section. To perform the tuning operation, sample a colour region on the image, select a Fine Tuning slider and move the slider to achieve the desired effect. See the **Fine Tuning Sliders** tool descriptions further on in this section for more details on slider selection.

Spill or L-poly Slider (Spill Removal)

When in the **Fine Tuning Actions** mode, this **Spill** slider can be used to remove spill from the registered colour region. After choosing the **Fine Tuning Actions** mode and registering a colour region, this slider can be moved to remove spill from the registered colour region. The more to the right the slider moves, the more spill will be removed. The more to the left the slider moves, the closer the colour component of the selected region will be to the colour in the original foreground image. If moving the slider all the way to the right does not remove all the spill, re-sample the colour region and again move the slider to the right. These slider operations are additive. This result achieved by moving the slider to the right can also be achieved

by clicking on the colour region using the **Spill(-)** operational mode. This slider bulges the Primatte large polyhedron near the registered colour region.

Transparency or M-poly Slider (Adjust Transparency)

When in the **Fine Tuning Actions** mode, this **Transparency** slider can be used to make the matte more translucent in the registered colour region. After choosing the **Fine Tuning Actions** mode and selected a colour region, moving this slider to the right makes the registered colour region more transparent. Moving the slider to the left makes the matte more opaque. If moving the slider all the way to the right does not make the colour region translucent enough, re-sample the colour region and again move the slider to the right. These slider operations are additive. This result achieved by moving the slider to the right can also be achieved by clicking on the colour region using the **Matte(-)** operational mode. This slider bulges the Primatte medium polyhedron near the registered colour region.

Detail or S-poly Slider (Add/Restore Lost Detail)

When in the **Fine Tuning Actions** mode, this **Detail** slider can be used to restore lost detail. After choosing the **Fine Tuning Actions** mode and selected a colour region, moving this slider to the left makes the registered colour region more visible. Moving the slider to the left makes the colour region less visible. If moving the slider all the way to the left does not make the colour region visible enough, re-sample the colour region and again move the slider to the left. These slider operations are additive. This result achieved by moving the slider to the left can also be achieved by clicking on the colour region using the **Detail(-)** operational mode. This shrinks the small polyhedron (which contains all the blue or green background colours) and releases pixels that were close to the background colour. The **S-poly** slider in the **Fine Tuning** mode is useful for restoring pixels that were lost because they were so similar to the background colour. This slider dents the Primatte small polyhedron near the registered colour region.

Spill Process Section

Complement/Solid/Defocus Spill Replacement

Allows the user to choose between the three methods of colour spill replacement as covered in detail in **Section 5. Spill Replacement Options** and below.

no suppression

Replaces the spill colour with the complement of the backing screen colour.

complement

Replaces the spill colour with the complement of the backing screen colour.

solid colour

Replaces the spill colour with colours from a defocused version of the background image.

defocused background

Replaces the spill colour with a 'user selected' solid colour.

Replace colour Slider

When **solid colour** is selected, this area allows the user to select a solid colour to use to replace the spill. For all other spill replacement selections, this area is 'greyed out' and not activated.

Defocus Slider

When **defocused background** is selected, this area allows the user to adjust the amount of defocus applied to the background buffer image. For all other spill replacement selections, this area is 'greyed out' and not activated.

Output Section

Output Mode Selector

These are the three formats for the output of the node.

composite

outputs the composite result of the Primatte node.

premultiplied

outputs the premultiplied result of the Primatte node.

unpremultiplied

outputs the unpremultiplied result of the Primatte node.

The Primatte Algorithm

There are three Primatte algorithms. Here is a chart that shows the main differences between them.

	Primatte	Primatte RT Plus	Primatte RT
Number of Separating Surfaces	128 (one for each color vector)	6	1
Saturated FG Support	OK	Not Supported	Not Supported
Color Suppression Model	Replacement/ Complement	Replacement	Replacement
Pixel Calculation Cost	Heavy	Light	Very Light

For a description of the Primatte algorithm, see *Explanation of how Primatte works* below.

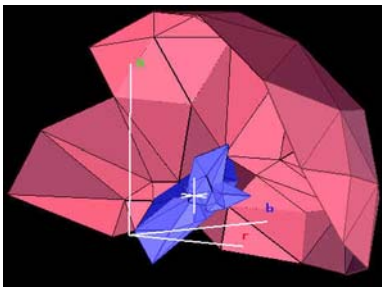
For a description of the Primatte RT+ algorithm, go to *Explanation of how Primatte RT+ works* on page 217.

For a description of the Primatte RT algorithm see *Explanation of how Primatte RT works* on page 218.

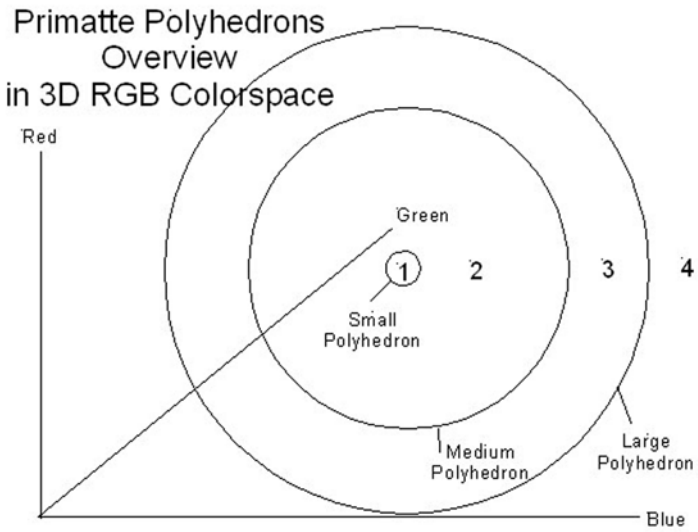
Explanation of how Primatte works

The Primatte chromakey algorithm is a sophisticated method of colour space segmentation that can be easily explained to help a user achieve maximum effectiveness with the tool. Basically Primatte segments all the colours in the foreground image into one of four separate categories. The result is a 'spill suppressed' foreground image and a matte which is used to apply the modified foreground to a suitable background.

Primatte works in 3D RGB colour space. Here is a visual representation of the Primatte algorithm after an image has been processed.



By operating the Primatte interface, the user essentially creates three concentric, multi-faceted polyhedrons. These can be pictured as three globes (or polyhedrons or polys), one within the other, which share a common centre point. The creation of these polyhedrons separates all possible foreground colours into one of four regions; inside the small polyhedron **(1)**, between the small and medium polyhedrons **(2)**, between the medium and the large polyhedrons **(3)** and outside the large polyhedron **(4)**.

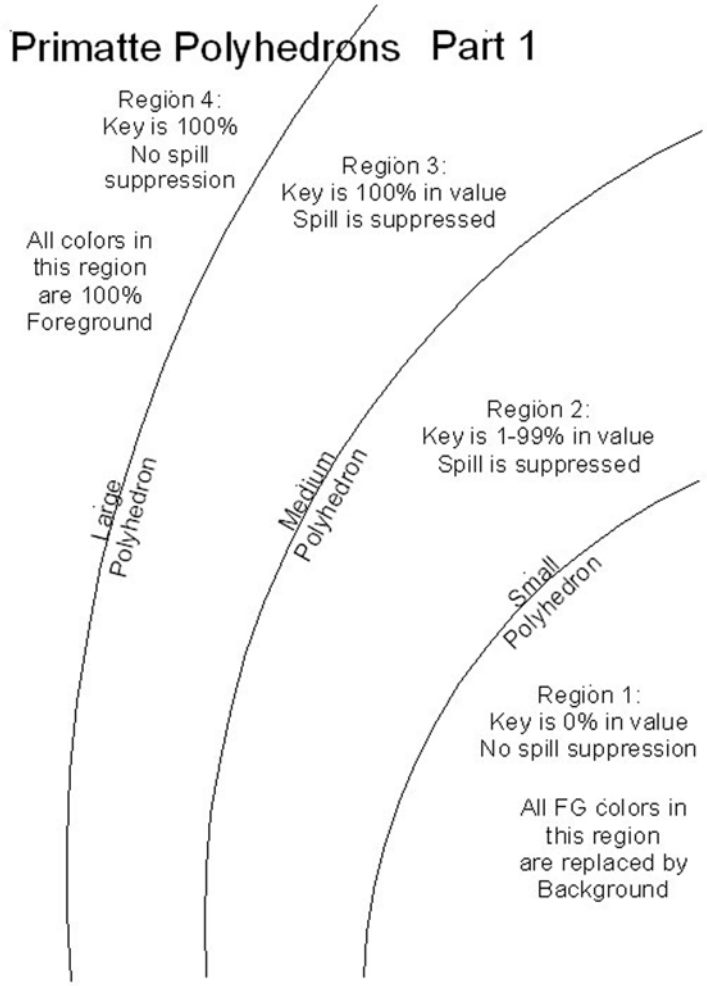


The four regions created are described as follows:

Region 1 (inside the small polyhedron) - This region contains all of the foreground image colours that are considered 100% background. These are the green or blue or whatever colours that were used as the backing colour of the foreground image.

Region 2 (between the small and medium polyhedrons) - This region contains all the foreground colours that are at the edges of the foreground object(s), in glass, glass reflections, shadows, sheets of water and other transparent and semi-transparent colour regions. These colour regions also have spill suppression applied to them to remove colour spill from the backing screen.

Region 3 (between the medium and large polyhedrons) - This region contains all the foreground image colours that are 100% foreground but have spill suppression applied to them to remove colour spill from the backing screen. Otherwise they are 100% solid foreground colours.



Region 4 (outside the large polyhedron) - This region contains all the 100% foreground image colours that are not modified from the original foreground image. There is no spill suppression applied to these colours.

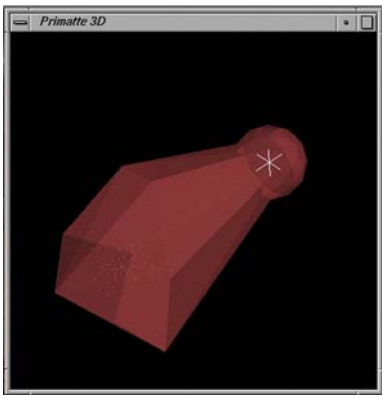
In the first step in using Primatte (**Select Background Colour**), the user is asked to indicate the backing colour on the original foreground image. The sample should usually be taken from a 'medium shaded' area near the foreground object. By 'medium shaded' area, it is meant that if green is the backing colour and the green area of the foreground image has many shades of green ranging from very pale green to almost black, a shade of green in-between these extreme ranges should be chosen. If good results are not obtained using this sample, Primatte should be reset and another sample taken using a slightly darker or lighter shade of green. The first sample of Primatte often determines the final result as the centre point of all three poly-hedrons is created based on this first sample.

A single pixel may be selected or a range of pixels (snail trail or rectangular sample). If a range of pixels is taken, the sample will be averaged to get a single colour sample. This single pixel or averaged colour sample then becomes the centre of the small polyhedron. A few other shades around that colour are included in the original small polyhedron.

Note

It is recommended that a single pixel be selected as the first sample as you then have some idea where the centre point of the polyhedrons is located. If a box sample or a long snail trail sample is made. You can only guess at the average colour that ends up being the centre point. You can get an idea how this sample affects the algorithm by resetting the Primatte plug-in, going to the Matte View and clicking around on the green or blue screen area while in the Select Background Colour Operation Mode. You can immediately see the results of the initial settings of the polyhedrons in this way.

After making a sample of the backing screen colour in the first step, the result is a small golf ball-shaped poly as shown in the following image.



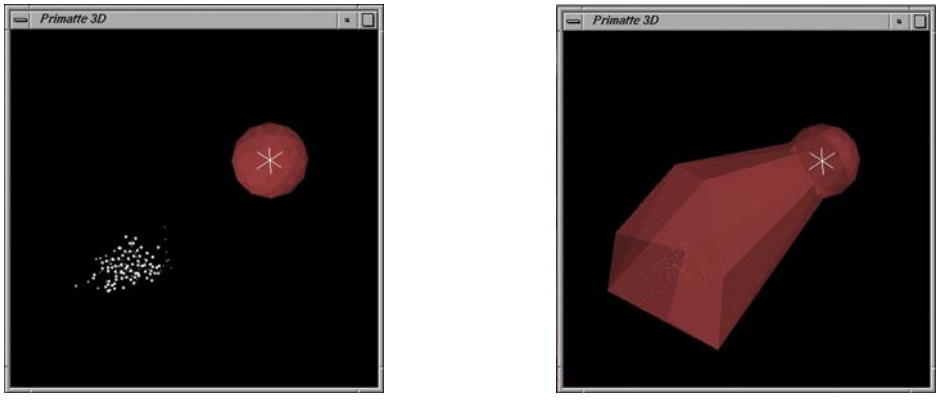
The second step in using Primatte is to clean up the backing colour area by adding additional shades of green or blue to the small poly. This second step (**Clean Background Noise**) is usually executed while viewing the black and white **Matte View**.

Before *After*

Figure 7-19: Before and after background noise removal.

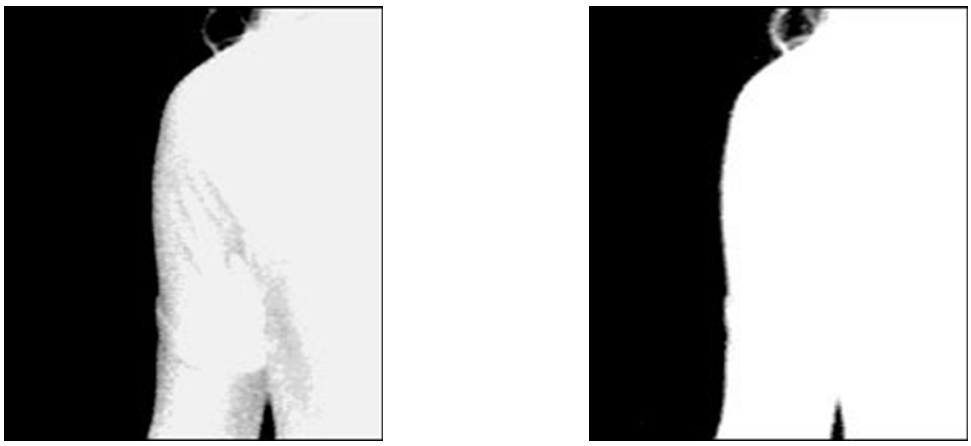
While in the **Clean Bg Noise** sampling mode, the user samples the white milky regions as shown in the left-hand image above. As the user samples these regions, they turn to black as shown in the right-hand image above.

What is happening in the Primatte algorithm is that these new shades of green (the white milky areas) are added to the small poly where all the shades of green or blue are moved. The next two images show the new pixels sampled (white dots) in relation to the small poly and the image next to it shows how the small poly extends outward to encompass the newly sampled colours into the small poly.



The advantage of this technique is that the polyhedron distorts to enclose only the shades of green that are in the backing screen. Other shades of green around these colours are left undisturbed in the foreground. Other chromakeyers expand from a golf ball-sized shape to a baseball to a basketball to a beach ball. Since it expands in all directions, many shades of green are relegated to 100% background making it hard to get good edges around the foreground objects.

Now that the user has created a small polyhedron, he must shape the medium and large polys. A default medium and large poly are both automatically created and are then modified based on the next couple of Primatte operations. The third Primatte step (**Clean Foreground Noise**) is to sample and eliminate gray areas in the 100% foreground area of the image.



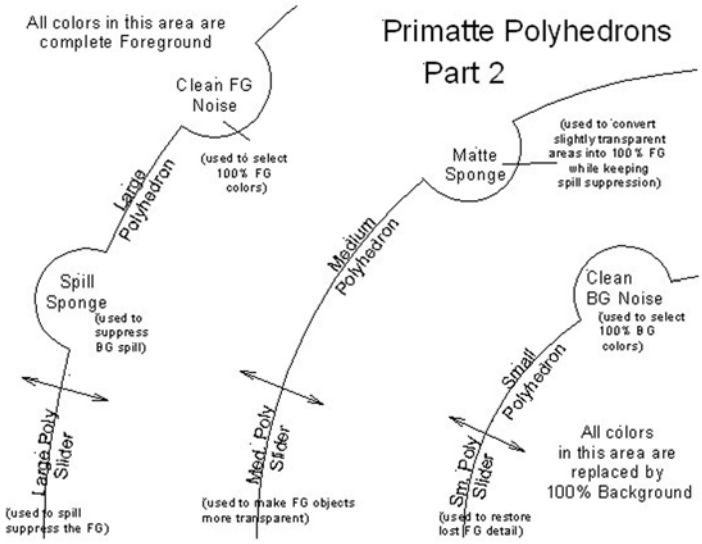
Before *After*

Figure 7-20: Before and after foreground noise removal.

Again, the user makes several samples on the dark, greyish areas on the foreground object until it is solid white in colour. Primatte is shaping the large polyhedron with each colour region that is sampled. Care should be taken in both this and the previous steps to not sample too close to the edges of the foreground object. Getting too close to the foreground object's edges will result in hard edges around the foreground object. Primatte uses these samples to modify and shape the medium and large polys to the desired shape. At this point, the matte or key has been created and would allow the foreground objects to be composited into a new background image.

If the user changes the display mode from the black and white **Matte View** to the colour **Composite View**, there is usually 'colour spill' on the edges (and sometimes the centre) of the foreground objects. When on the edges of the foreground object, this spill comes from where the edges of the foreground object blended into the backing colour. If it is on the centre of the foreground object, it usually results from reflected colour from the backing screen. The next Primatte step, either **Spill Sponge**, **Fine Tuning** or **Spill(-)**, can now be used to eliminate this spill colour.

Let's take a look at what is happening in the Primatte algorithm while this next step is performed. Here is what the various tools in Primatte do to the Polyhedrons when they are used:


Primatte Polyhedrons
Part 2
All colors in this area are complete Foreground
Clean FG Noise
(used to select 100% FG colors)
Large Polyhedron
Spill Sponge
(used to suppress BG spill)
Large Poly Slider
(used to spill suppress the FG)
Matte Sponge
(used to convert slightly transparent areas into 100% FG while keeping spill suppression)
Medium Polyhedron
Med. Poly Slider
(used to make FG objects more transparent)
Clean BG Noise
(used to select 100% BG colors)
Small Polyhedron
Sm. Poly Slider
(used to restore lost FG detail)
All colors in this area are replaced by 100% Background

As you can see above, the **Spill Sponge** bulges the large polyhedron in the colour region specified. A colour region is specified by clicking on the image in a particular area with spill present. For example, if the user clicks on some spill on the cheek of a foreground person, Primatte goes to the section of the large polyhedron closest to that particular flesh tone and bulges the polyhedron there. As a result, the flesh tones move from outside the large poly to in-between the medium and large polys. This is **Region 3** and, if you remember, is 100% foreground with spill suppression. As a result of the suppression, the spill is removed from that cheek colour and all other shades of that colour on the foreground. The user would then continue to sample areas of the image where spill exists and each sample would remove spill from another colour region.

When all spill has been removed, the user should have a final composite. As a last step, he should go back to the **Matte View** and make sure that gray, transparent areas have not appeared in the foreground area. If there are any, the **Matte Sponge Operation Mode** should be selected and those gray pixels are sampled until they have all turned white again.

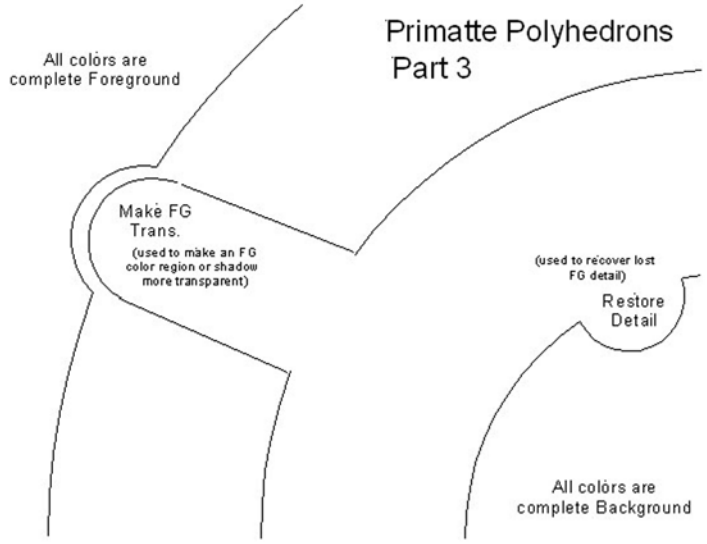
The **Matte Sponge** and **Spill Sponge** tools bulge or dent the polyhedrons a pre-selected amount. If the desired results are not achieved or the results are too extreme for the image, a manual method can be applied. The user should choose the **Fine Tuning** sliders, select a colour region of interest and then move the appropriate slider to get the desired results.
For example, to remove spill, select a region of the composite image with spill on it. Move the spill or large poly slider to the right a little bit, the large poly will bulge and the spill should disappear. Move it a little more, if necessary. Moving this slider to the right removes spill (moves the colours from outside the large poly to between the medium and large polyhedrons) and moving it to the left, dents the large poly and moves that colour region to outside the large poly.

If the user samples a foreground object shadow and then moves the **Matte** or medium poly slider to the right, the shadow will become more transparent. This is useful for matching com-

posited shadows to shadows on the plate photography. It can also be used to make clouds or smoke more transparent.

If some foreground detail disappears during the composite, the user can select where the detail should be and then move the detail or small poly slider to the left. This dents the small poly in that colour region and releases the detail pixels from the small poly into the visible region between the small and medium polyhedrons.

The **Spill Sponge** and **Matte Sponge** tools are 'shortcut tools' that automatically move the sliders a pre-selected amount as a timesaving step for the user. Other 'shortcut tools' include the **Make Foreground Trans.** tool and the **Restore Detail** tool.



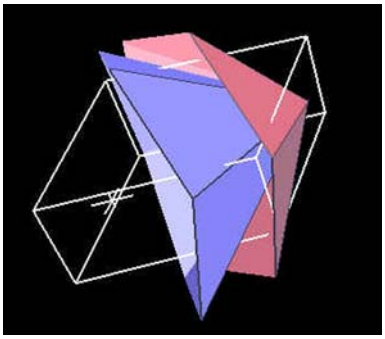
These 'shortcut tools' are one-step operations where the user clicks on a colour region of interest and Primatte performs a pre-calculated operation. Hopefully, most operations using Primatte would only require these tools, but the manual operation of the sliders is always an option.

The **Spill(-)** tool bulges the large poly a small amount incrementally in the colour region that is clicked on and the **Spill(+)** tool dents it a small amount with each click. The **Matte(-)** and **Matte(+)** tools do the same to the medium poly and the **Detail(-)** and **Detail(+)** do it to the small poly.

Explanation of how Primatte RT+ works

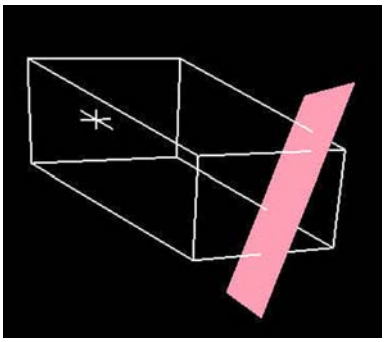
The **Primatte RT+** algorithm differs from the Primatte algorithm in that it has a six surface colour separator instead of the 127-faceted polyhedrons. This makes the **Primatte RT+** algorithm much simpler and, therefore, faster to calculate. The results and performance of **Primatte RT+** falls in between the **Primatte** and **Primatte RT** options. Where the **Primatte RT+** algorithm might not work well is with less saturated backing screen colours and it also does not support the **Complement Colour** spill suppression method (which is the spill suppression method that delivers the best detail). For a well-lit and photographed image or clip, this algorithm will produce good results and render quickly.

Here is what a visual representation of the Primatte RT algorithm looks like after an image has been processed:



Explanation of how Primatte RT works

Primatte RT is the simplest algorithm and, therefore, the fastest. It uses only a single planar surface to separate the 3D RGB colorspace and, as a result, does not have the ability to separate out the foreground from the backing screen as carefully as the above Primatte algorithm. Like the **Primatte RT+** algorithm, **Primatte RT** might not work well with less saturated backing screen colours and it too does not support the **Complement Colour** spill suppression method (which is the spill suppression method that delivers the best detail). For a well-lit and photographed image or clip, this algorithm will produce good results and render very quickly.
Here is what a visual representation of the **Primatte RT** algorithm looks like after an image has been processed:



Contact Details

Main office

IMAGICA Corp. of America, 1840 Century Park East, #750, Los Angeles, CA, USA 90067

Primatte office

IMAGICA Corp. of America, 3113 Woodleigh Lane, Cameron Park, CA 95682. Phone: 1-530-677-9980, FAX: 1-530-677-9981, Cell: 1-530-613-3212, Email: sgross@imagica-la.com, Web Site: http://primatte.com

Proprietary Notices

Primatte is distributed and licensed by IMAGICA Corp. of America, Los Angeles, CA, USA. Primatte was developed by IMAGICA Corp., Tokyo, Japan. Primatte is a trademark of IMAGICA Corp., Tokyo, Japan.

8 Using Paint

Nuke features a vector-based Paint node for help with tasks like rig removal and dustbusting. This chapter gives full instructions on its usage.

Connecting the Paint Node

The Paint node accepts a single foreground input and up to three background inputs. It requires only the foreground input and uses the optional background inputs with the Reveal tool (see *Using the Reveal Tool* on page 221).

To connect the Paint node:

1. Click **Draw** > **Paint** to add a new Paint node.
2. Drag the unnamed input to the node whose output you wish to use as the foreground.
3. If you plan to reveal pixels from a background element, drag the **bg** input to the node whose output you wish to use.
4. Repeat the above as necessary with the **bg2** and **bg3** inputs.

Applying Strokes

Any given Paint node can hold many strokes, and you apply them using any of the following tools.

Icon	Tool	Function
	Freehand	Applies colours atop the current plate.
	Reveal	Applies pixels from a source plate to a destination plate.
	Eraser	Removes pixels from existing paint strokes.
	Clone	Applies pixels from one region of the current plate to another region of the current plate.

A separate Select tool lets you make changes to a stroke once it's been drawn (see *Selecting Strokes for Editing* on page 223).

This section discusses the general steps for using each of these tools, and gives instructions on editing the attributes, timing, and stack order of strokes.

Using the Freehand Tool

The Freehand tool lets you apply coloured strokes to the current plate.

Figure 8-1: Painting with the Freehand tool.

To use the Freehand tool:

1. Set colour, opacity, brush size, brush hardness, and comp mode as described below (see *Editing Stroke Attributes* on page 223).
2. Set the timing of the stroke as described below (see *Editing Stroke Timing* on page 227).
3. Click the Freehand tool.
4. Apply strokes as necessary.

Using the Reveal Tool

The Reveal tool lets you pull pixels from background elements onto the foreground element. You can pull pixels from directly beneath the pointer, or you can introduce an offset in order to pull from a region adjacent to the pointer.

The Reveal tool requires at least one background input (see *Connecting the Paint Node* on page 220); otherwise, your strokes will draw in black.

Figure 8-2: Painting with the Reveal tool.

To use the Reveal tool:

1. Set opacity, brush size, brush hardness and comp mode as described below (see *Editing Stroke Attributes* on page 223).
2. Set the timing of the stroke as described below (see Editing Stroke Timing on page 241).

3. Click the Reveal tool.
4. Set the **source** parameter to the input you want to pull pixels from.
5. If you wish to reveal pixels that are offset from the pointer, **Ctrl**/**Cmd**+drag from the source to the destination to set the offset. Alternatively, you can enter the offset numerically using the **offset** parameter.
6. If you want to make sure the copied pixels are not filtered, check **round?**. If this is not checked and you turn on the proxy mode, even an integer offset may be scaled to a non-integer.
7. Start painting. The pointer overlay depicts the source of the offset as a crosshair within a circle and the destination as a circle (the diameter of which represents the breadth of the stroke).
8. Apply strokes as necessary.

Using the Eraser Tool

The Eraser tool lets you remove pixels from existing paint strokes.

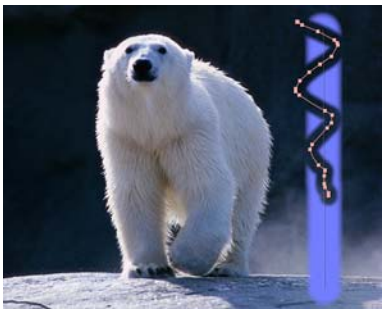


Figure 8-3: Painting with the Eraser tool.

To use the Eraser tool:

1. Set opacity, brush size, and brush hardness as described below (see *Editing Stroke Attributes* on page 223).
2. Set the timing of the stroke as described below (see Editing Stroke Timing on page 241).
3. Click the Eraser tool.
4. Apply strokes as necessary.

Using the Clone Tool

The Clone tool lets you remove unwanted features from the plate by painting over them with pixels offset from the pointer.

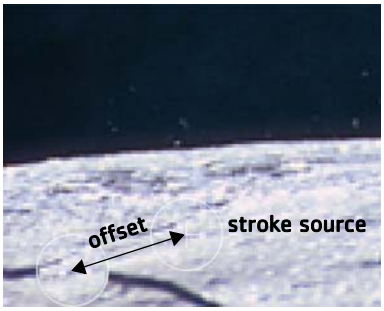


Figure 8-4: Painting with the Clone tool.

To use the Clone tool:

1. Set opacity, brush size, brush hardness and comp mode as described below (see *Editing Stroke Attributes* on page 223).
2. Set the timing of the stroke as described below (see Editing Stroke Timing on page 241).
3. Click the Clone tool.

4. To set the offset, hold down **Ctrl/Cmd** and left-click at the source location, drag to where you want to paint, and release. Alternatively, you can enter the offset numerically using the **offset** parameter. If you'd like the offset amount to be rounded to an integer, check **round?**.
5. Start painting. The pointer overlay depicts the source of the offset as a crosshair within a circle and the destination as a circle (the diameter of which represents the breadth of the stroke).
6. Apply strokes as necessary.

Selecting Strokes for Editing

If you've already applied a stroke but wish to make changes to it, you can select it with the Select tool.

To select a stroke:

1. Click the Select tool.

2. Select the stroke you wish to edit either by clicking on it in the Viewer or by clicking on its name in the stroke list. To select several strokes, **Ctrl/Cmd**+click on their names in the stroke list.

Editing Stroke Attributes

Prior to applying a stroke, or after selecting a stroke with the Select tool, you can edit the following attributes, each of which is animateable. The attributes you edit prior to applying a stroke will vary according to the tool you plan to use.

Editing Colour

When using the Freehand tool (see *Using the Freehand Tool* on page 221), you can set the RGBA colour values of the stroke using Nuke's standard colour controls (see Chapter 2, *Using the Colour Picker and Colour Controls,* on page 51).

***Color** set to white (the default).*

***Color** set to light blue.*

Figure 8-5: Editing the **color** attribute.

Editing Opacity

You can set the opacity of the stroke using the **opacity** slider. You can also temporarily make the stroke invisible (that is, completely transparent) by checking the **hide** box.

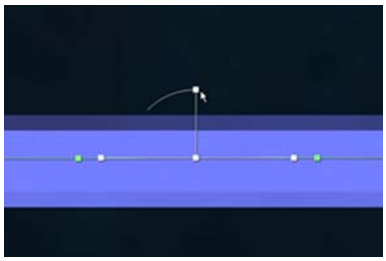
*A low **opacity** value.*

*A high **opacity** value.*

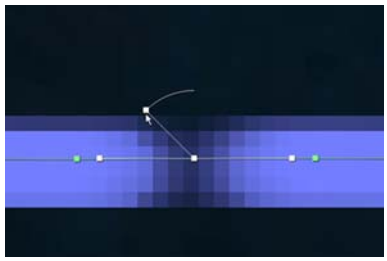
Figure 8-6: Editing the **opacity** attribute.

Finally, if you are using a tablet, you can tie a stroke's transparency to pen pressure. Just check the **opacity** box. If you also check **edit pressure**, you can move the handles that

appear in the Viewer to alter whatever is tied to the pen pressure (**opacity**, **size**, and/or brush **hardness**).



Original paint stroke.

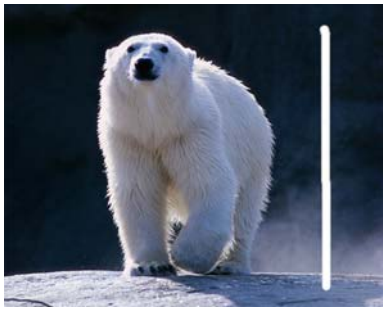


Adjusting opacity.

Figure 8-7: Altering the opacity of the paint stroke with the Viewer handles instead of pen pressure. To do this, you need to check **opacity** and **edit pressure**.

Editing Brush Size

You can set the size of the stroke using the **brush size** slider.

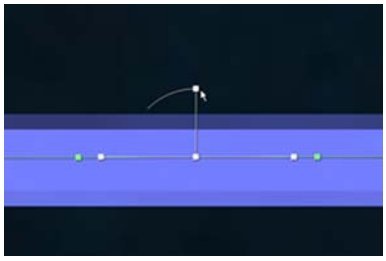


A low brush size value.

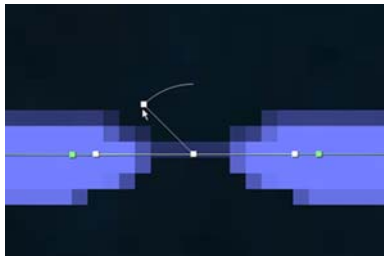
A high brush size value.

Figure 8-8: Editing the **brush size** attribute.

If you are using a tablet, you can also tie a stroke's size to pen pressure by checking the **size** box. If you also check **edit pressure**, you can move the handles that appear in the Viewer to alter whatever is tied to the pen pressure (**opacity**, **size**, and/or brush **hardness**).



Original paint stroke.



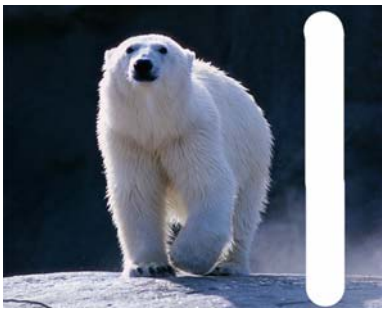
Adjusting size.

Figure 8-9: Altering the size of the paint stroke with the Viewer handles instead of pen pressure. To do this, you need to check **size** and **edit pressure**.

Editing Brush Hardness

You can set the hardness of the stroke using the **brush hardness** slider.

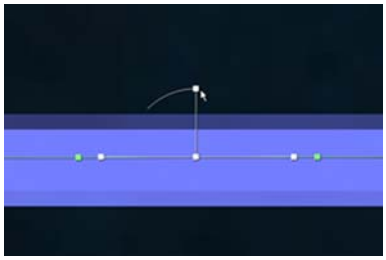
*A low **brush hardness** value.*



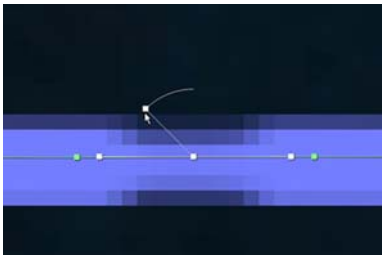
*A high **brush hardness** value.*

Figure 8-10: Editing the **brush hardness** attribute.

If you are using a tablet, you can also tie a stroke's hardness to pen pressure by checking the **hardness** box. If you also check **edit pressure**, you can move the handles that appear in the Viewer to alter whatever is tied to the pen pressure (**opacity**, **size**, and or brush **hardness**).



Original paint stroke.



Adjusting brush hardness.

Figure 8-11: Altering the hardness of the paint stroke with the Viewer handles instead of pen pressure. To do this, you need to check **hardness** and **edit pressure**.

Editing Comp Mode

Using the **Comp Mode** parameter, you can control which components of the image are affected by the paint stroke. Options include the following:

- **draw** - This is the default option. The paint stroke affects all the channels of the image.
- **luma** - The paint stroke only affects the luminance of the image.
- **chroma** - The paint stroke only affects the hue and saturation of the image.

- **alpha** - The paint stroke only affects the alpha channel. This can be useful for cleaning up a matte, for example. Note that when an alpha channel does not exist in the input, the Paint node does not create one.

Comp mode set to *luma.*

Comp mode set to *chroma.*

Figure 8-12: Editing the **comp mode** attribute.

Editing Stroke Timing

Prior to applying a stroke, or when altering a stroke with the Select tool (see *Selecting Strokes for Editing* on page 223), you can edit the range of frames during which a stroke is visible.

To make stroke visible for all frames (the default):

Press the Unlimited Lifetime button.

To make stroke visible from the current to end frame:

Press the Display from Now to End button.

To make stroke visible only on current frame:

Press the Display on Current Frame Only button.

To make stroke visible from the beginning to current frame:

Press the Display from Start to Now button.

To make stroke visible during a specified range for frames:

1. Press the Display for Set Range button. A dialog box prompts for the frame range.
2. Enter the start and end frames for the range during which you wish the stroke to appear and click **OK**.

Editing Stroke Stack Order

Prior to applying a stroke, or when altering a stroke with the Select tool (see *Selecting Strokes for Editing* on page 223), you can edit its foreground to background drawing order.

To move a stroke to the background (behind all strokes):

Press the Move to Background button.

To move a stroke one step closer to the background:

Press the Move Back button.

To move stroke one step closer to the foreground:

Press the Move Forward button.

To move a stroke to the foreground (in front of all strokes):

Press the Move to Foreground button.

Editing Stroke Vectors

To edit a stroke vector, you first need to activate the Select tool and select the stroke in the Viewer or the stroke list. You can then modify the points that make up the stroke vector.

To add a point to a vector:

1. With the Select tool active, select the paint stroke in the Viewer or the stroke list.
2. In the Viewer, select any one of the points that make up the paint stroke.
3. **Ctrl/Cmd+Alt**+click on the spot where you want to add a new point.

To move a point:

1. With the Select tool active, select the paint stroke in the Viewer or the stroke list.
2. In the Viewer, select the point that you want to move.
3. Drag the point to a new location. To constrain the point's movement to the x or y axis only, hold down **Ctrl/Cmd+Shift** when dragging the point. To adjust the curve slope around the point, drag the point's tangents to a new location.

To move several points together:

1. With the Select tool active, select the paint stroke in the Viewer or the stroke list.
2. In the Viewer, drag a marquee around the points (or an entire paint stroke) that you want to move. A transform overlay appears.
3. Adjust the transform overlay as necessary (see *Using the 2D Transformation Overlay* on page 159).

To close and open vectors:

1. With the Select tool active, select the paint stroke in the Viewer or the stroke list.
2. In the Viewer, right-click on any one of the points that make up the paint stroke.
3. Select **open/close curve**.

If the curve is open, Nuke connects the first point in the vector to the last point. If the curve is closed, the first and the last points in the vector become disconnected.

To delete a point:

1. With the Select tool active, select the paint stroke in the Viewer or the stroke list.

2. In the Viewer, select the point that you want to delete.
3. Press **Delete**.

OR

1. With the Select tool active, select the paint stroke in the Viewer or the stroke list.
2. In the Viewer, right-click on the point that you want to delete.
3. Select **delete point**.

To delete an entire paint stroke:

1. In the Paint controls, click on the stroke in the stroke list to select it.
2. Click **delete**.

OR

1. In the Paint controls, activate the Select tool.
2. In the Viewer, click on the stroke.
3. Right-click on one of the points that make up the stroke.
4. Select **delete curve**.

OR

1. In the Paint controls, activate the Select tool.
2. In the Viewer, click on the stroke and select one of the points that make up the stroke.

Press **Ctrl/Cmd+X**.

Animating Paint Strokes

All paint strokes that appear on more than one frame can be animated.

To animate a paint stroke:

1. Draw a paint stroke that appears on more than one frame.
2. Activate the Select tool and select the paint stroke from the list in the Paint node controls.

3. Move to a new frame.
4. Adjust the stroke vector or its attributes as necessary. A key frame is automatically set. The frame marker on the timeline turns cyan to indicate the selected paint stroke is animated.

   If you made changes to a paint stroke vector, the vector overlay in the Viewer also turns from pink to light green to indicate that a key frame has been set.
5. Repeat steps 3 and 4 for all the frames you want to set as key frames.

To delete a key frame:

1. Using the Viewer timeline, scrub to the frame where you want to delete a key frame.
2. In the Paint node controls, select the paint stroke whose key you want to delete.
3. Do one of the following:

- If you want to delete a key set to animate a vector, click the **cut** button under **curve keys.** The vector overlay (which you can see in the Viewer whenever the Paint node's Select tool is active) turns from light green to pink to indicate a key isn't set at the current frame. If no attribute keys are set for the selected paint stroke either, the frame marker on the timeline turns from cyan to the default colour.
- If you want to delete a key set to animate attributes, click the **cut** button under **attribute keys**. Nuke deletes the attribute key, but the frame marker on the timeline may not turn from cyan to the default colour. This is because keys may still be set to animate the shape of the selected paint stroke.

Copying, Pasting and Cutting Stroke Attributes

After creating a paint stroke, you can copy, paste, and cut its attributes (such as colour and brush size) to use them in other paint strokes.

Copying Attributes

The **copy** button under **attribute keys** lets you copy the attributes of selected strokes at the current frame to the clipboard. Unlike the **cut** button described below, it does not delete any keys set.

To copy attributes:

1. In the Viewer, scrub to the frame that contains the paint stroke(s) whose attributes you want to copy.
2. Select the paint stroke(s) from the list in the Paint node controls.
3. Under **attribute keys**, click **copy**.

   Nuke copies all the attributes of the selected paint strokes to the clipboard. Any keys set to animate these attributes are not affected.

Pasting Attributes

The **paste** button under **attribute keys** lets you paste any attributes that are on the clipboard to the selected strokes at the current frame. It also set keys at the current frame to animate these attributes. The shape of the strokes or any keys set to animate their shape are not affected.

To paste attributes:

1. In the Viewer, scrub to the frame that contains the paint stroke(s) to which you want to apply the attributes on the clipboard.
2. Select the paint stroke(s) from the list in the Paint node controls.
3. Under **attribute keys**, click **paste**.

   Nuke applies the attributes on the clipboard to the selected paint strokes and sets the current frame as a key frame. The attributes of the selected paint strokes on other frames are not affected.

Cutting Attributes

The **cut** button under **attribute keys** lets you delete any keys set at the current frame to animate the attributes of selected strokes. It also copies these attributes to the clipboard. The shape of the strokes or any keys set to animate their shape are not affected.

To cut attributes:

1. In the Viewer, scrub to the frame that contains the paint stroke(s) whose attributes you want to cut.
2. Select the paint stroke(s) from the list in the Paint node controls.
3. Under **attribute keys**, click **cut**.

   Nuke deletes any keys set to animate the attributes of the selected paint strokes, and copies all these attributes to the clipboard.

Copying, Pasting and Cutting Stroke Vectors

After creating a paint stroke, you can copy, paste, and cut its vector to use the same shape in other paint strokes.

Copying Vectors

The **copy** button under **curve keys** lets you copy the vectors of selected strokes at the current frame to the clipboard. Unlike the **cut** button described below, it does not delete any keys set.

To copy vectors:

1. In the Viewer, scrub to the frame that contains the paint stroke(s) whose vectors you want to copy.
2. Select the paint stroke(s) from the list in the Paint node controls.
3. Under **curve keys**, click **copy**.

   Nuke copies the vectors of the selected paint strokes to the clipboard. Any keys set to animate these vectors are not affected.

Pasting Vectors

The **paste** button under **curve keys** lets you paste any vectors that are on the clipboard to the selected strokes at the current frame. It also set keys at the current frame to animate these vectors. The attributes of the strokes or any keys set to animate the attributes are not affected.

To paste vectors:

1. In the Viewer, scrub to the frame that contains the paint stroke(s) to which you want to paste the vectors on the clipboard.
2. Select the paint stroke(s) from the list in the Paint node controls.
3. Under **curve keys**, click **paste**.

   Nuke pastes the vectors (but not any attributes) on the clipboard to the selected paint strokes and sets the current frame as a key frame. The vectors of the selected paint strokes on other frames are not affected.

Cutting Vectors

The **cut** button under **curve keys** lets you delete any keys set at the current frame to animate the vectors of selected strokes; however, it does not delete the vectors themselves. Cut also copies the vectors at the current frame to the clipboard. The paint stroke attributes or any keys set to animate the attributes are not affected.

To cut vectors:

1. In the Viewer, scrub to the frame that contains the paint stroke(s) whose vectors you want to cut.
2. Select the paint stroke(s) from the list in the Paint node controls.
3. Under **curve keys**, click **cut**.

   Nuke deletes any keys set to animate the vectors of the selected paint strokes, and copies the vectors to the clipboard.

Paint and Stereoscopic Projects

When you are using the Paint node in a stereoscopic or multi-view project, you can use the **view** control to select the view to paint on. For more information, see *Selecting the View to Process When Using the Paint Node* on page 336.

Using the **correlate** controls on the Paint node, you can create a paint stroke on one view and have it automatically generated for the other. For more information, refer to *Reproducing Paint Strokes* on page 338.

9 Temporal Operations

This chapter explains the temporal or time-based operations in Nuke. You learn how to distort time (that is, slow down, speed up, or reverse clips), apply motion blur, and perform editorial operations like slips, cuts and splices. This chapter also discusses techniques for working with interlaced video footage and preparing it for use in a compositing project.

Distorting Time

Time distortion changes the length of time required to playback a clip in your composite. These operations generally fall under one of two categories: *retiming* and *warping*.

Retiming is the process of slowing playback by adding frames, or accelerating playback by subtracting frames.

Warping is the process of slowing down, speeding up, or even reversing playback on a clip without necessarily altering the overall length.

Tip

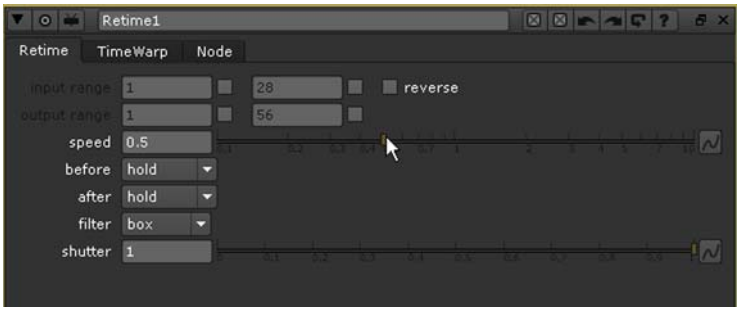
When working with temporal operations, it helps to attach a Viewer to the retiming or warping node so you can see the effect of your changes.

Simple Retiming

Nuke's Retime node lets you change the playback time for all the frames in a clip or for range of frames within the clip. You can also use it to reverse the clip playback. It does this by dropping or duplicating frames. For higher quality retiming see *OFlow Retiming* on page 237.

To retime all frames in a clip:

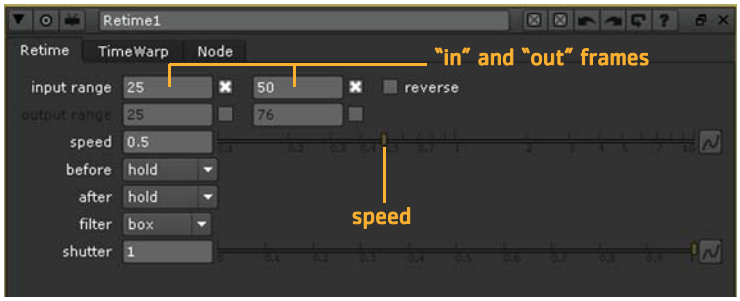
1. Choose **Time** > **Retime** to insert a Retime node into your script.
2. Enter a value in the **speed** parameter. Values higher than 1 increase playback speed; values less than 1 decrease playback speed.



3. Check the **reverse** box if you want to play the clip backwards—making the last frame the first, the first frame the last, and so on.
4. Increase the **shutter** parameter to enable frame-blending (For more information, see *Interpolation* below).

To retime a range of frames in a clip:

1. Choose **Time > Retime** to insert a Retime node into your script.
2. Check the boxes for **input range** and enter the "in" and "out" frames.



   For example, if your original clip is 50 frames, but you want to only retime the last half, you would input **25** for the in point and leave the out point at **50**.
3. Check the box for **output range** and enter the "in" and "out" frames to retime to a specific clip length.

   or

   Enter a factor in the **speed** parameter and Nuke will calculate the **output range** for you. Values higher than 1 increase playback speed; values less than 1 decrease playback speed.
4. Check the **reverse** box to invert the selected frame range.
5. Increase the **shutter** parameter to enable frame-blending.

Interpolation

Time distortions that slow down a clip require the creation of additional, or *interpolated*, frames. For example, suppose you want to slow down the collision, shown in the clip below, by a factor of two. This requires the creation of one interpolated frame for every existing frame in the clip.

Figure 9-1: Interpolating frames.

The simplest way for Nuke to interpolate is to duplicate existing frames and increase the length of the clip—this is the default method of interpolation. However, this method can create jittery playback, especially when the image depicts very fast motion and the clip is retimed to be considerably longer than its original length. For such cases, Nuke provides different nodes for smoothly interpolating the frames.

Frame-blending

The Frame Blend node interpolates frames by generating an additive composite of the frames that precede and follow it, rather than creating mere copies between the existing frames. Here is an example of the difference between frame-copy and frame-blend interpolation. In the first frame of Figure 9-2, you see a copy of the preceding frame. In the second frame, you see a new image generated by blending the previous and subsequent frames.

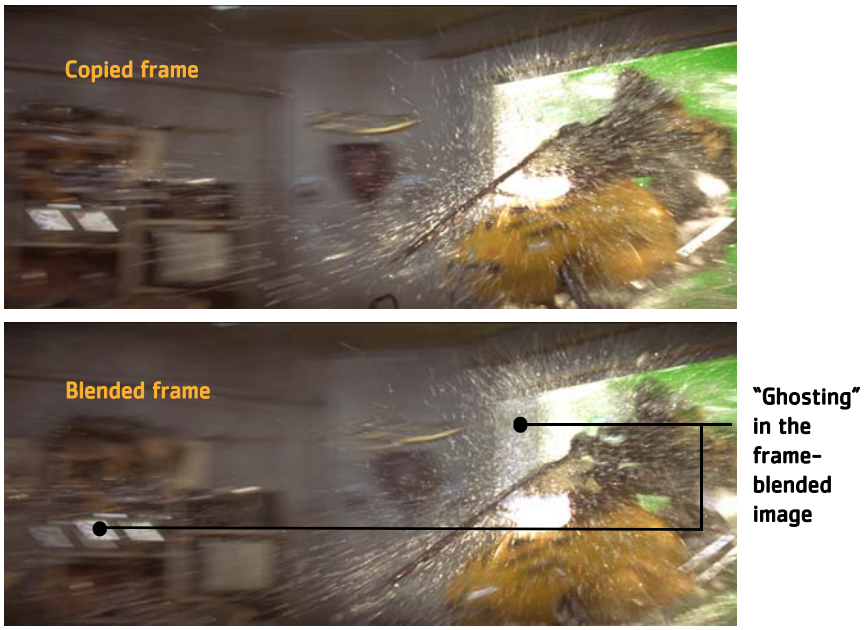


Figure 9-2: Frame-copied versus frame-blended interpolation.

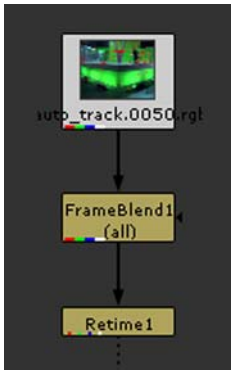
The latter method creates "ghosting" around all fast moving features (the window frame and the pages on the desk, for example). This may look odd when viewed as part of a still frame, but will contribute to smoother motion during actual playback.

You can enable frame-blending by manipulating the **shutter** value of a retiming node. Higher shutter values generate more frame-blending. Or, you can insert a Frame Blend node before the temporal effect you want to influence. The below figure shows an example of Frame Blend with the Retime node.

To insert a FrameBlend node:

1. Choose **Time > FrameBlend** from the menu.

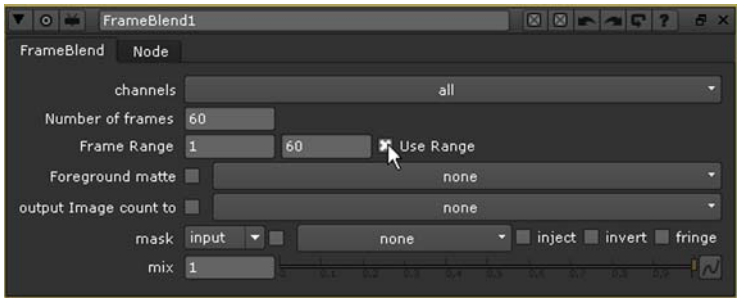
   Remember to place it upstream from the temporal effect you want to influence.



2. Enter the number of frames to blend in the **Number of frames** field.

   or

Check the box for **Use Range** and enter the starting and ending frames that you want to blend



3. If necessary, check **Foreground matte** and select the channel to limit the blending effect.

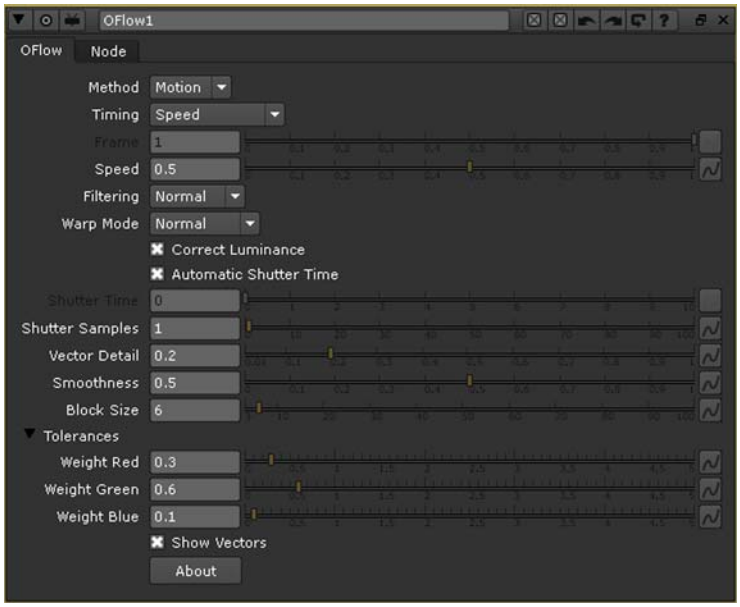
The option beneath Foreground matte (**output Image count to**) saves a floating point alpha image to a channel you specify; the result indicates the number of images that contributed to each pixel of the matte. To normalize the alpha, divide the number 1 by the number of frames averaged, and then multiply the alpha channel by this result. You can also use the inverse of this matte for additional degraining.

OFlow Retiming

The OFlow node generates high-quality retiming operations analysing the movement of all pixels in the frames and then rendering new "in-between" images based on that analysis. This node can also add motion blur or enhance the existing motion blur in the image.

To retime with OFlow:

1. Select the node for the clip that you want to retime.
2. Choose **Time** > **OFlow** from the menu bar.
3. Set the speed of the output clip. A value of 0.5 will slow the movement down.



That's pretty much it.

If you prefer you can map input to output frames to retime the clip. For example, if you wanted to halve the speed of a 50 frame clip using this method, switch Timing to Source Frame. On frame 1 set a key for the Frame value to be 1. On frame 50 set a key for the Frame value to be 25.

OFlow Parameters

The following table describes the different parameters in the OFlow node's controls.

OFlow Parameter	Function
Method	Sets the interpolation algorithm. • **Frame** - the nearest original frame is displayed. • **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and is useful when tweaking the timing on a curve before setting the method to motion. • **Motion** - vector interpolation is used to calculate the in-between frame.
Timing	Sets how to control the new timing of the clip. • **Speed** - select this if you wish to describe the retiming in terms of "double speed" or "half speed". • **Source Frame** - select this if you wish to describe the retiming in terms of "at frame 100 in the output clip I want to see frame 50 of the source clip". You'll need to set at least 2 key frames for this method to work.
Frame	This parameter is active only if timing is set to Frame. Use this to specify the source frame at the current frame in the time bar. For example, to slow down a 50 frame clip by half set the Frame to 1 at frame 1 and the Frame to 50 at frame 100. The resulting animation curve will result in a half-speed retime.

OFlow Parameter	Function
Speed	This parameter is only active if Timing is set to Speed. Values below 1 slow down the clip. Values above 1 speed up movement. For example, to slow down the clip by a factor of 2 (half speed) set this value to 0.5. Quarter speed would be 0.25.
Filtering	Sets the quality of the filtering when producing in-between frames. • **Normal** - uses bilinear interpolation which gives good results and is a lot quicker than extreme. • **Extreme** - uses a sinc interpolation filter to give a sharper picture but takes a lot longer to render.
Warp Mode	Sets how to control the new timing of the clip. • **Simple** - this is the quickest option, but may produce poor results around moving objects and image edges. • **Normal** - this is the default option with better treatment of moving objects and image edges. • **Occlusions** - this is the advanced option which attempts to reduce the level of background dragging that occurs between foreground and background objects.
Correct Luminance	Local motion estimation is highly dependent upon the idea that the brightness of objects don't vary through a sequence. Where brightness varies rapidly - for example a highlight moving across the bodywork of a car - the motion calculation will perform poorly. The luminance of a shot can come from other sources too - such as an overall flicker problem. In these cases where there is a global luminance shift, toggling this control on will allow the local motion estimation algorithm to take account of overall brightness changes between frames.
Automatic Shutter Time	Calculates the shutter time throughout the sequence automatically.
Shutter Time	Sets the equivalent Shutter Time of the retimed sequence. A shutter time of 1 is equivalent to averaging over plus and minus half an input frame which is equivalent to a shutter angle of 360 degrees. A shutter time of 0.5 is equivalent to a shutter angle of 180 degrees. Imagine a grey rectangle moving left to right horizontally across the screen. The figures below show how Shutter Time affects the retimed rectangle. **Shutter Time 1** **Shutter Time 0.5**

OFlow Parameter	Function
Shutter Samples	Sets the number of in-between images used to create an output image during the shutter time. Increase this value for smoother motion blur, but note that it takes much longer to render. **Shutter Samples 2** **Shutter Samples 20**
Vector Detail	Adjust this to vary the resolution of the vector field. Large vector fields take longer to process, but contain more detail and may help to isolate smaller motion in the scene. A value of 1 will generate a vector for every pixel. A value of 0.5 will generate a vector at every other pixel. For some sequences, a high vector detail near 1.0 generates too much unwanted local motion detail and often a low value will give a better result.
Smoothness	Vector fields usually have two important qualities: they should accurately match similar pixels in one image to another and they should be smooth rather than noisy. Often it is necessary to trade one of these qualities off against the other. A high **smoothness** will miss lots of local detail, but is less likely to provide you with the odd spurious vector. A low **smoothness** will concentrate on detail matching, even if the resulting field is jagged. The default value of 0.5 should work well for most sequences.
Block Size	The vector generation algorithm subdivides the image into small blocks, and separately tracks them. **blockSize** defines the width and height of these subdivisions. Smaller values will produce noisy data, whereas larger values may produce data that is lacking in detail. This value should rarely need editing; some sequences may benefit from using large block sizes to help the algorithm track regions better where the algorithm isn't "locking on" to the overall motion in the sequence.
Tolerances	For efficiency, much of the local motion estimation is done on luminance only - i.e. using monochrome images. The tolerances parameters allow you to tune the weight of each colour channel when calculating the image luminance. These parameters rarely need tuning. However, you may, for example, wish to increase the red weighting **Weight Red** to allow the algorithm to concentrate on getting the motion of a primarily red object correct, at the cost of the rest of the items in a shot.
Weight Red	The red weighting used when calculating the vector field.
Weight Green	The green weighting used when calculating the vector field.
Weight Blue	The blue weighting used when calculating the vector field.

OFlow Parameter	Function
Show Vectors	Switch this on to draw the motion vectors over the image.
About	Shows the version number of this node.

Tip

Old-time Nuke users may remember nodes called OpticalFlow and OptiFlow. OFlow replaces these nodes. It can be used for retiming and adding motion blur, but it does not have the capability to output motion vectors. To output motion vectors, you could use F_VectorGenerator (included in The Foundry's Furnace plug-ins).

Warping Clips

Warping refers to slowing down, speeding up, or even reversing select frames in a clip without necessarily altering its overall length. Otherwise stated, warps add, subtract, or reverse the temporal detail in a range of frames within a clip. For example, Figure 9-3 depicts a snowmobile clip (downsampled to just ten frames for easy representation) that we might want to warp.

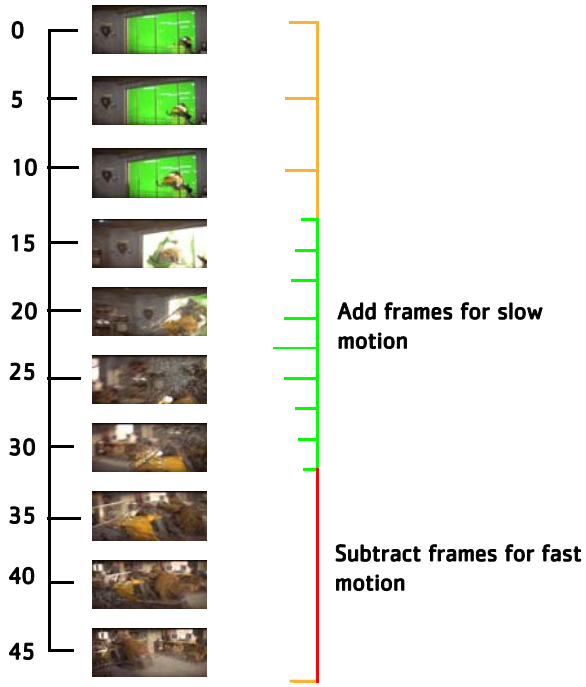


Figure 9-3: Planning the time warp.

One way—in fact, kind of the classic way—to warp this clip would be to play the frames just prior to the collision at their original rate, the frames involving the collision in slow motion, and the frames after the collision in fast motion.

You could achieve such a warp by sculpting the curve in Nuke's TimeWarp curve, which is a part of the Retime node's parameters, to look something like the one depicted below.

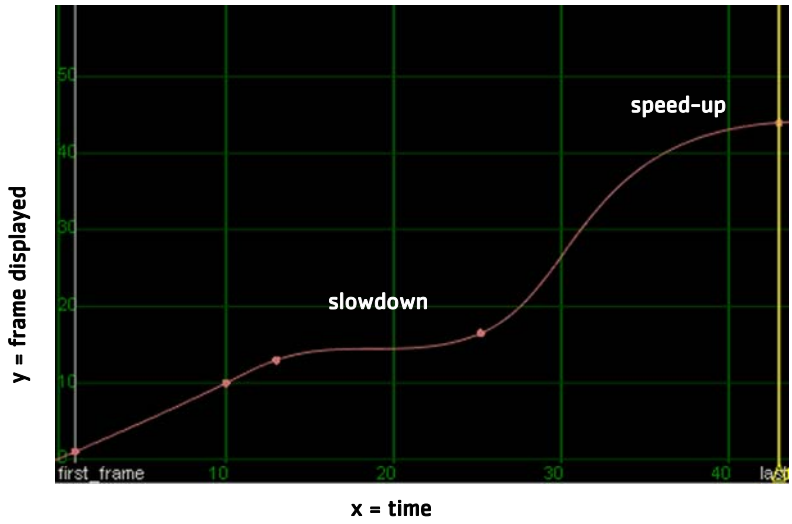


Figure 9-4: Editing the warp curve.

The basic "rules" for editing the warp curve are as follows:

- To slow down motion, decrease the slope of the curve.
- To speed up motion, increase the slope of the curve.
- To reverse motion, create a downward sloping portion on the curve (a dip, in other words).

To warp a clip:

1. Click **Time > Retime** to insert a Retime node into your script.
2. Click the **TimeWarp** tab to reveal the TimeWarp curve.
3. Attach a Viewer to this node, so you can see the effect of your changes.
4. Sculpt the TimeWarp curve according to the rule above. (**Ctrl/Cmd+Alt** click to insert key frame knots on the curve; **Ctrl/Cmd**+drag to reposition key frame knots; **Ctrl/Cmd**+drag to rotate a key frame knot control handles.)
5. If you want to enable frame blending on the output, either input a value larger than one in the Retime node's **shutter** parameter, or insert a FrameBlend node prior to the Retime node.

Global Frame Range and Speed

Nuke automatically adjusts the timeline of every Viewer window you open to show the "in" and "out" frames for the clip you're viewing.

After you retime a clip in your compositing script, you may need to adjust the script's global frame range and playback speed (frames-per-second), to account for the retiming operations.

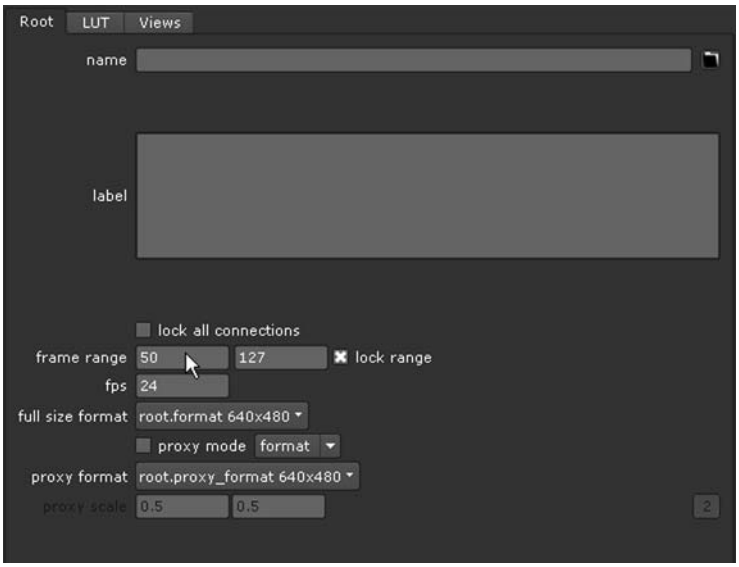


Figure 9-5: Adjusting the global frame range.

Choose **Edit > Project settings** (or press **S** over the Nuke window) and then enter the new **frame range** and **fps** in the Project settings properties panel.

Applying the TimeBlur Filter

When a fast moving subject is recorded on film or video, its edges appear to smear as a result of the object's movement while the shutter is open. The longer the shutter remains open at each frame interval, the more obvious this effect. Time Blur simulates this phenomenon by sampling its input at "divisions" times over "shutter" frames starting at the current frame plus "offset".

Time blur is commonly applied to garbage masks that are tracked to a fast moving feature. The time blur averages the incoming mask image over the shutter period, better match the motion blur in the original image and creating a more convincing integration.

To apply motion blur to a clip:

1. Click **Time** > **TimeBlur** to insert a TimeBlur node into your script. Place it downstream from the element to which you want to apply motion blur.
2. In the TimeBlur properties panel, set **divisions** to the number of times you want to sample the input over the shutter time. For images with fast-moving content higher values will be necessary to eliminate "steppiness" or banding in the output.
3. Set **shutter** to equal the span of time (in frames) over which the input should be sampled. A shutter time of .5 is typical and would correspond with a camera shutter of 180 degrees.
4. Set the **shutteroffset** to control when the sampling of the input starts relative to the frame being rendered, analogous to when the camera shutter opened to capture corresponding film or video footage you might have at the same frame. You may need to adjust this by eye to align, for example, a garbage mask with an underlying feature.

Editing Clips

As a node-based system, Nuke doesn't have a timeline. Nevertheless, you can still perform editorial operations that you might associate with a timeline. You can slip clips (move them forward or backward in time), cut them, or splice them to other clips.

Slipping Clips

Slipping a clip refers to moving it backward or forward in time. There are any number of reasons why you might want to do this (for example, to synchronize events in a background and foreground clip).

To slip a clip:

1. Click **Time > TimeOffset** to insert a TimeOffset node into your script. (Place it downstream from the element to which you want to slip.)
2. Attach a Viewer to this node, so you can see the effect of your changes.
3. In the TimeOffset properties panel, check **reverse input** if you want to invert the clip (make the last frame the first, and so on).
4. In the **time offset (frames)** field, type the number of frames by which you want to slip the clip. Enter a negative value to subtract frames from the head of the clip. Enter a positive value to add frames to the head of the clip.
5. Adjust the script length for the new output range. Choose **Edit** > **Project settings**, and enter **frame range** values that match the output range you specified.

Note

It's not mandatory that you adjust the script's frame range after slipping the clip. If you don't, the Viewer will fill the empty frames at the tail of the clip by holding on the last frame.

Cutting Clips

Cutting a clip refers to shortening it by removing frames from its head or tail.

To cut a clip:

1. Click **Time** > **FrameRange** to insert a FrameRange node into your script. Insert it downstream from the element to which you want to cut.
2. Attach a Viewer to this node, so you can see the effect of your changes.
3. In the **frame range** fields, then enter the appropriate in and out point values.

   For example, if your original clip is 50 frames but you want to use only the last 25 frames in your composite, you would enter **25** as the In point and leave the Out point at **50**.

4. Adjust the script length for the new output range. Choose **Edit** > **Project settings**, and enter **frame range** values that match the output range you specified.

Note

It's not mandatory that you adjust the script's frame range after cutting the clip. If you don't, the Viewer will simply fill the empty frames at the head or tail of the clip by holding on the first or last frame.

Splicing Clips

Splicing refers to joining clips head-to-tail, thus allowing action to flow from one shot to the next. When you splice clips, you have options for:

- Fading to or from black.
- Dissolving from the first to second clip.
- Slipping the combined clip in time.

To splice clips:

1. Click **Time** > **AppendClip** to insert an AppendClip node into your script.
2. Attach its **1** and **2** pipes to the clips you want to join. (The clip attached to pipe 1 will precede the one attached to pipe 2.)
3. Attach a Viewer to this node, so you can see the effect of your changes.
4. If necessary, expand the script length to accommodate the total length of the newly merged clip:
   - Click **Edit > Project settings**. The Project settings properties panel appears.
   - Enter **frame range** values that matches the total length.
5. In the **Fade In** and **Fade Out** fields of the AppendClip properties panel, type the number of frames, if any, you want to fade to or from black.

   For example, typing a **5** in the **Fade In** field would result in the following effect at the head of the merged clip.

   (The inverse of this effect would occur at the *tail* of the merged clip were you type **5** in the **Fade Out** field.)
6. In the **Cross Dissolve** field, type the number of frames, if any, of overlap you want between the first and second clip.

   For example, leaving **Cross Dissolve** at the default of **0** creates a simple cut—the transition from the first to second clip is instantaneous. Typing in a **5** creates a time span

of five frames in which the first clip's gain ramps downward to zero, while the second's ramps upward to 100%.

Figure 9-6: Dissolve

Figure 9-7: Cut

7. In the **First Frame** field, type the number of frames, if any, by which you want to slip the clip. Enter a negative value to subtract frames from the head of the merged clip. Enter a positive value to add frames to the head of the clip.
8. Slipping the merged clips may create empty black frames at its head or tail. As appropriate, choose **First frame** or **Last frame** if you want these empty frames to appear as copies of the first or last frame.

10 Warping and Morphing Images

Nuke's warping and morphing tools allow you to distort elements in an image, apply and correct lens distortions, and morph one image into another. The nodes designed for these operations include the GridWarp node, the SplineWarp node, and the iDistort node. In this chapter, we focus on the GridWarp and SplineWarp nodes.

Warping

Warping refers to manipulating an image so that elements in the image are distorted. Unlike many of the transformations described under *Transforming Elements* on page 159, warps are transformations that only affect some of the pixels in an image rather than all of them. For example, you might make an animal's eyes bigger or a person's smile wider without affecting the rest of their features.

This is not to say that the pixels around the area you are moving do not move with the area. They do, because accommodating the change this way often produces more realistic results. However, the distortion lessens the further you get from the moved pixels. You also have some control over which pixels are moved and which are not, and can isolate the warp to a small area. Still, in an ideal situation, the subject you are going to warp is a subject you can key out or rotoscope to isolate it from its background before you create the warp. This way, you can be sure that the background stays intact.

In addition to performing creative manipulations on the shapes of the subjects in your images, you can also use warping to simulate different types of film or video lenses or to remove unwanted lens distortions.

Below, we discuss how to warp images, first using the GridWarp node and then the SplineWarp node. Finally, we also teach you to animate the warps. Again, we start with the GridWarp node and then show you how to do the same with the SplineWarp node.

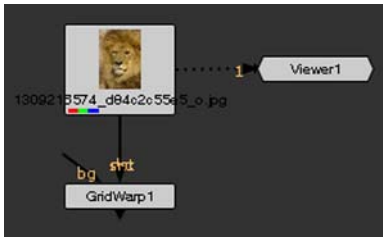
Warping Images Using the GridWarp Node

The GridWarp node allows you to warp images by transferring image information from one bezier grid onto another. When using this node, you first create the source grid, which defines where to warp from. Next, you create the destination grid, which defines where to warp the image to. This grid can be a duplicate of the source grid, or you can draw it separately. When you manipulate the destination grid, the corresponding warp is applied to the source image.

The GridWarp node also includes controls for animating the warp, adding motion blur to warps that occur quickly, and selecting the level of antialiasing used to remove any artifacts the warp may have caused.

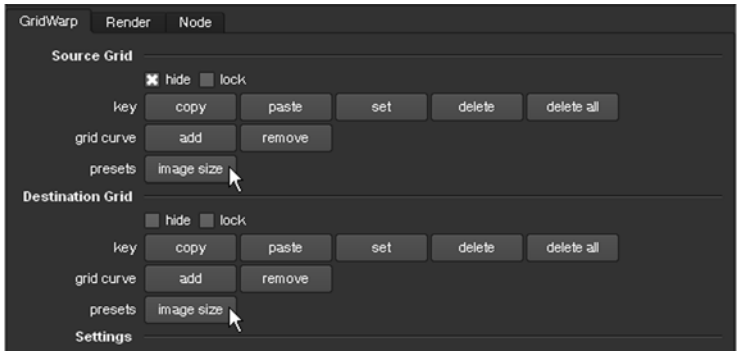
To warp an image using the GridWarp node:

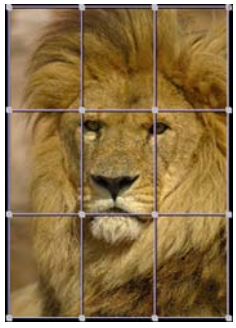
1. Select **Transform** > **GridWarp** to insert a GridWarp node after the image you want to warp.
2. Connect both the **src** and the **dst** input and a Viewer to the image.



3. When the GridWarp properties panel is open, you can see the source and destination grids appear as small overlays in the Viewer. The source grid is pink, and the destination grid blue. In the following steps, you use the pink source grid to define which areas you want to warp and the blue destination grid to define where to warp these areas to.

4. To make the grids the same size as the input image, click the **image size** buttons under both **Source Grid presets** and **Destination Grid preset.**

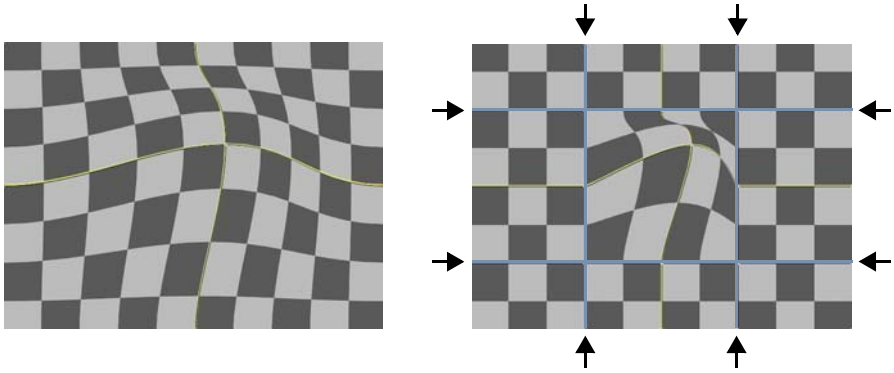




5. For now, check **hide** under **Destination Grid** to hide the blue destination grid in the Viewer. This way, you can't accidentally distort the image yet.
6. Modify the grid around the area you want to warp. Usually, you want the grid to conform to the subject of the source image. For example, if you are warping an animal's eyes, you need to create grid lines that follow the edges of the eyes.

   To add more points and lines to the grid, click the **add** button under the **Source Grid** controls and click on an existing grid line in the Viewer. If you click on a horizontal line, a vertical line is added to the grid. If you click on a vertical line, a horizontal line is added to the grid. The lines can be further apart in the areas that you don't intend to warp.

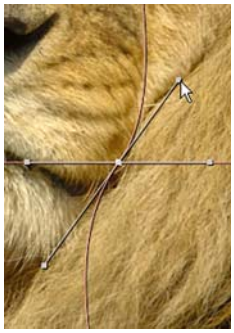
   You can use the grid lines to isolate the areas you do not want to warp. You do this by adding lines between the area you intend to warp and the area you don't want to change.



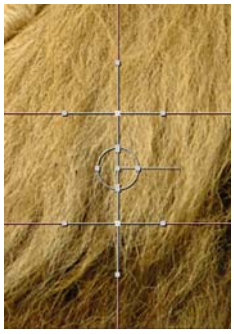
An unlimited warp

The same warp limited to a small area with grid lines

When you select a point, four tangent handles appear around it. You can use these handles to modify the curves connecting the points.



To move several points together, draw a marquee around them and use the transformation overlay that appears. For more information on how to use the transformation overlay, see *Using the 2D Transformation Overlay* on page 159.



To remove a line, click the **remove** button and the line you want to remove.

7. If necessary, animate the source grid to match any movement in the source image. For more information on how to do this, see *Animating Warps* on page 255.
8. Unless you want to draw the destination grid separately, click **copy** under **Source Grid** and **paste** under **Destination Grid**. This copies the source grid you created in the previous step into the destination grid.
9. Connect the Viewer to the GridWarp node.

10. Hide the source grid (check **hide**) and lock it to prevent accidental changes (check **lock**). Show the destination grid instead (uncheck **hide**). It should look the same as the source grid, only blue (assuming you copied the source grid into the destination grid).

11. In the areas where you want to warp the image, drag the points on the grid to a new position. The pixels in these areas are moved in the direction you moved the points. Pixels in the nearby areas are also moved to accommodate the change, but the distortion lessens the further you get from the repositioned points. If you don't want a nearby area distorted, add more grid lines between the area and the points you want to move before you drag the points to a new location.

12. To better see what the warped image looks like, press **O** on the Viewer to toggle the overlay off. To compare the original and warped images, press **D** repeatedly on the GridWarp node to disable and enable it. If you see changes in the areas you don't want to warp, go back to modifying the destination grid.

13. If necessary, animate the destination grid to match any movement in the destination image.

14. If necessary, adjust the controls described in the following table.

Control	What it does
reverse	Check this to invert the distortion.
submesh resolution	Set the number of subdivisions that are created between bezier curves in the grid. The higher the value, the more accurate the distortion between the grid lines.
distortion	The overall distortion is multiplied by this amount. The lower the value, the less the image is distorted. At 0, you get the source image, while at 1 you get the destination image. At 0.5, the distortion is halfway between the source and the destination images.
blend	Dissolve between the source image (at 0) and the destination image (at 1).
background	The warped image is rendered on top of an unwarped background. This control sets what to use as that background: • **on black** - Render the warped image on top of a constant black image. • **on src** - Render the warped image on top of the image connected to the **src** input of the GridWarp node. • **on dst** - Render the warped image on top of the image connected to the **dst** input of the GridWarp node. • **on bg** - Render the warped image on top of a background image connected to the **bg** input of the GridWarp node.
background blend	Blend between the output of the GridWarp node (at 0) and whatever you have selected from the **background** pulldown menu (at 1).
filter (on the **Render** tab)	Choose the appropriate filtering algorithm (see *Choosing a Filtering Algorithm* on page 160).
antialiasing (on the **Render** tab)	Select the level of antialiasing to reduce any aliasing artifacts the warp may have caused.
samples (on the **Render** tab)	Enter the number of samples to render per pixel when motion blurring. The higher the number, the smoother the result. Setting the value to 0 produces no motion blur.
shutter (on the **Render** tab)	Enter the number of frames the shutter stays open when motion blurring. For example, a value of 0.5 would correspond to half a frame. Increasing the value produces more blur, and decreasing the value less.

Control	What it does
shutteroffset (on the **Render** tab)	Select when the shutter opens and closes in relation to the current frame value when motion blurring: • to centre the shutter around the current frame, select **centred**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter will stay open from frame 29,5 to 30,5. • to open the shutter at the current frame, select **start**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter will stay open from frame 30 to 31. • to close the shutter at the current frame, select **end**. For example, if you set the **shutter** value to 1 and your current frame is 30, the shutter will stay open from frame 29 to 30. • to open the shutter at the time you specify, select **custom**. In the field next to the pulldown menu, enter a value (in frames) you want to add to the current frame. To open the shutter before the current frame, enter a negative value. For example, a value of - 0.5 would open the shutter half a frame before the current frame.
randomize time (on the **Render** tab)	Adjust this to add randomness to the distribution of samples in time so they don't produce regularly spaced images. The larger the value, the larger the time difference between the samples.
sample diameter (on the **Render** tab)	Adjust the diameter of the circle that the samples for each pixel are placed in for antialiasing. The larger the value, the more pixels are jittered.

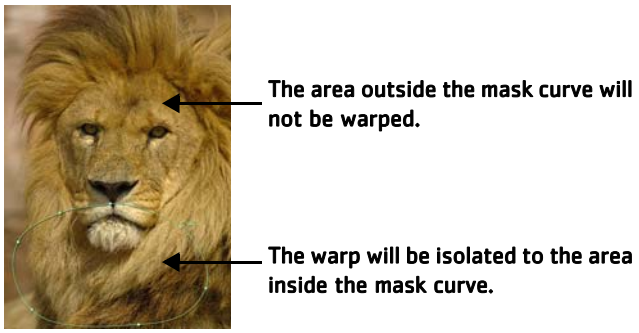
Warping an Image Using the SplineWarp Node

The SplineWarp node deforms an image based on multiple bezier curves that you create. The source curve defines where to warp from, while the destination curve defines where to warp the source image to. Unlike with the GridWarp node, you can draw these curves anywhere on the image rather than only add points on the existing grid lines. The controls for adding and modifying points are similar to the controls of the Bezier node.

To warp an image using the SplineWarp node:

1. Select **Transform** > **SplineWarp** to insert a SplineWarp node after the image you want to warp.
2. Connect both the **src** and the **dst** input to the image. Attach a Viewer to the SplineWarp node.

3. If you need to animate the warp, make sure **autokey** is checked in the SplineWarp node's controls. This way, the curves you will soon create will automatically be set as key shapes for the animation.
4. Make sure **show both curves** is not checked and **show** is set to **blend**.
5. If there are areas in the image that you do not want to warp, check **first bezier masks deformation**. Then, **Ctrl+Alt+click** (Mac users **Cmd+Alt+click**) on the image in the Viewer to draw a curve that isolates the area you intend to warp from the area you want to keep untouched.



When you select a point, two tangent handles appear around it. You can use these handles to modify the curves connecting the points.

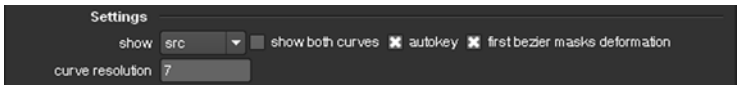
To move several points together, draw a marquee around them and use the transformation overlay that appears. For more information on how to use the transformation overlay, see *Using the 2D Transformation Overlay* on page 159.

To close an open curve or open a closed curve, right-click on a point and select **open/close curve**.

To remove a point, right-click on the point and select **delete point**.

To remove an entire curve, right-click on a point on the curve and select **delete curve**.

6. Set **show** to **src** so you can create the source curves that define where to warp from.

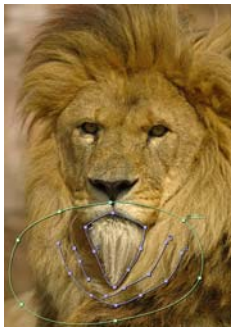


7. In the Viewer, **Ctrl+Alt+click** (Mac users **Cmd+Alt+click**) to create the pink source curves. Usually, you want the curves to conform to the subject of the source image. For example, if you are warping an animal's eyes, you need to create curves that follow the edges of the eyes.

   You probably want to create more than one curve around the area you are warping. To begin a new curve, make sure none of the existing points are selected and **Ctrl/Cmd+Alt+click** on the image. The curves are numbered in the order you create them.

8. If you need to animate the warp, scrub to another frame on the timeline and adjust the curves to match any movement in the source image. If **autokey** is checked, key shapes are set automatically. For more information on how to animate the warp, see *Animating Warps* on page 255.
9. Unless you want to create the destination curves separately, click **copy** under **Source Curves** and **paste** under **Destination Curves** to use the curves you just created as the basis for the destination curves.

10. Then, set **show** to **srcwarp**. You should now see the blue destination curves that are identical to the source curves (assuming you copied the source curves into destination curves).
11. In the areas where you want to warp the image, drag the points on the curves to a new position. The pixels in these areas are moved in the direction you moved the points.



12. To better see what the warped image looks like, press **O** on the Viewer to toggle the overlay off. To compare the original and warped images, press **D** repeatedly on the SplineWarp node to disable and enable it. If you see changes in the areas you don't want to warp, go back to modifying the mask or the destination curves.
13. If necessary, move to other frames and animate the destination grid to match any movement in the destination image.
14. If necessary, adjust the controls described in the following table.

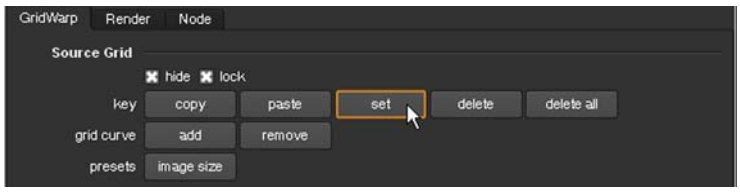
Control	What it does
show	Select the image to output: • **src** – the unwarped image from the **src** input. • **srcwarp** – the warped image from the **src** input. • **dst** – the unwarped image from the **dst** input. • **dstwarp** – the warped image from the **dst** input. • **blend** – a dissolve between the warped images from the **src** and **dst** inputs.
show both curves	Check this if you want to display both the source and destination curves in the Viewer at the same time.
curve resolution	Increase this value to improve the accuracy of how the warp matches the curves.
preview resolution	Increase this value to improve the quality of the OpenGL preview.
distortion	The overall distortion is multiplied by this amount. The lower the value, the less the image is distorted. At 0, you get the source image, while at 1 you get the destination image. At 0.5, the distortion is halfway between the source and the destination images. To see the effect, you need to set **show** to **blend**.
blend	Dissolve between the source image (at 0) and the destination image (at 1). To see the effect, you need to set **show** to **blend**.

Animating Warps

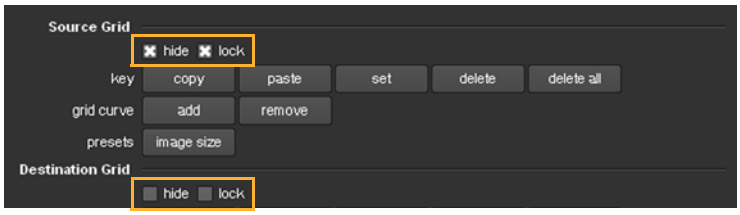
Unless you are warping a still image, you probably want to animate the warp. In the GridWarp and SplineWarp node's properties panels, there are controls for animating both the source and the destination grids or curves. Here, we first look at the GridWarp node and then the SplineWarp node. The instructions assume you know how to warp a still image using these nodes (if not, refer to *Warping Images Using the GridWarp Node* on page 247 and *Warping an Image Using the SplineWarp Node* on page 252).

To animate a warp using the GridWarp node:

1. While viewing the source image, adjust the pink source grid as necessary (see *To warp an image using the GridWarp node:* on page 247 for information on how to do this). When you are happy with the grid, click the **set** button under **Source Grid**. This saves the current grid as a key shape.



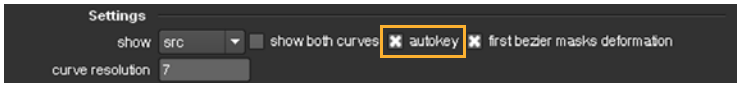
2. Move to a new frame and adjust the source grid accordingly. A new key shape is set automatically.
3. Repeat the previous step as necessary. If you need to delete a key shape, scrub to the frame where you set it and click **delete** under **Source Grid**. If you want to delete the entire animation and make the current grid the source grid for all frames, click **delete all**.
4. Click **copy** under **Source Grid** and **paste** under **Destination Grid** to copy the source grid into the destination grid.
5. Hide the pink source grid and display the blue destination grid.



6. While viewing the output of the GridWarp node, adjust the destination grid until you are happy with the warp. Click the **set** button under **Destination Grid**. The current grid is saved as a key shape.
7. Move to a new frame and adjust the destination grid again. The modified grid is automatically set as a key shape.
8. Repeat the previous step until you are happy with the animated warp.

To animate a warp using the SplineWarp node:

1. If you want the key shapes created automatically when you create the curves, make sure **autokey** is checked in the SplineWarp controls.



2. Create the source curves as instructed under *To warp an image using the SplineWarp node:* on page 252. If **autokey** is checked, the current curves are saved as key shapes. If autokey is not checked, you need to click the **set** button under **Source Curves** to set the curves as key shapes.

3. Scrub to a new frame on the timeline, and adjust the curves to conform to the movement of the image. If you have not checked autokey, click **set** under **Source Curves** again.
4. Repeat the previous step as necessary until you are happy with the source curve animation. If you need to delete a key shape, scrub to the frame where you set it and click **clear** under **Source Curves**. The current key shape is deleted and the curves for the frame are calculated by interpolation.
5. Click **copy** under **Source Curves** and **paste** under **Destination Curves** to copy the source curves into the destination curves.
6. Animate the destination curves in the same way as the source curves (only using the controls under **Destination Curves**).

Morphing

Morphing refers to dissolving two images together so that the subject of one image seems to change shape and turn into the subject of the other through a seamless transition. A morph can be easily noticeable or very subtle. An example of a noticeable morph would be a man turning into a woman or one animal turning into another, whereas a transition from an actor to his stunt man would result in a much more subtle morph.

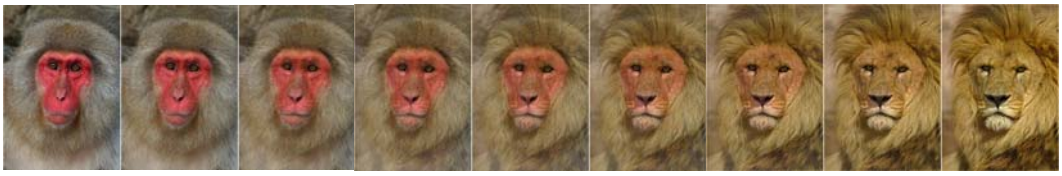


Figure 10-1: An image of a monkey turning into an image of a lion.

Morphing can be a time-consuming task, but it can be made easier by good advance planning of the shots. The more similar the characteristics, position, and movement of the subjects you want to morph are, the easier it is to morph them together. If the position and the movement of the subjects do not match, however, you can try to reposition and retime the clips before morphing them together. To do so, use the nodes described in Chapter 5, *Transforming Elements,* on page 159 and Chapter 9, *Temporal Operations,* on page 233. You can also use the Tracker node (see Chapter 6, *Tracking and Stabilising,* on page 175) to track the features you want to morph over time or to stabilize your clips before morphing them together.

Tip

You can copy animation data from the Tracker node to the points on the GridWarp/SplineWarp grid/curves. Do the following:

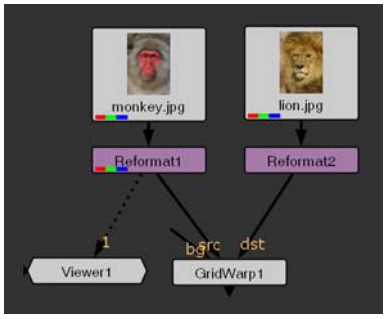
1. Make sure both the Tracker and the GridWarp/SplineWarp properties panels are open.

2. **Ctrl/Cmd**+drag the animation button from the Tracker node properties on top of the individual grid/curve point in the Viewer. When you release the mouse, the point follows the animation from the Tracker node.

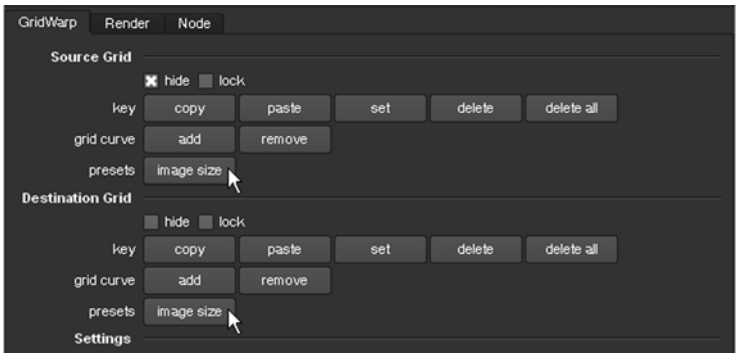
Below, we first discuss morphing images using the GridWarp node and then using the SplineWarp node.

To morph one image into another using the GridWarp node:

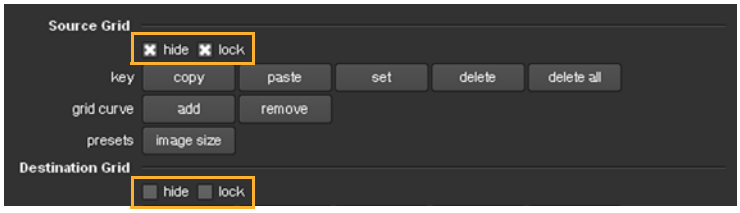
1. Select **Image** > **Read** to import the two images you want to morph together.
2. If the images do not have the same resolution, insert Reformat nodes after them. For more information, see *Reformatting Image Sequences* on page 99.
3. Select **Transform** > **GridWarp** to insert a GridWarp node into your script.
4. Connect the source image (the image you want to turn into another image) to the **src** input of the GridWarp node, and the destination image (the image you want the source image turned into) to the **dst** input. Connect a Viewer to the source image (or its Reformat node if one exists).



5. To make the grids the same size as the input images, click the **image size** buttons under both **Source Grid presets** and **Destination Grid presets**.



6. In the GridWarp controls, check **hide** under **Destination Grid**.
7. Identify some key features in the source image that correspond to features in the destination image, and adjust the pink source grid to conform to these features. For more information on how to adjust the grid, see *To warp an image using the GridWarp node:* on page 247.
8. Click **copy** under **Source Grid** and **paste** under **Destination Grid**. This copies the source grid you created in the previous step into the destination grid.
9. Uncheck **hide** under **Destination Grid**. Instead, check **hide** and **lock** under **Source Grid**.



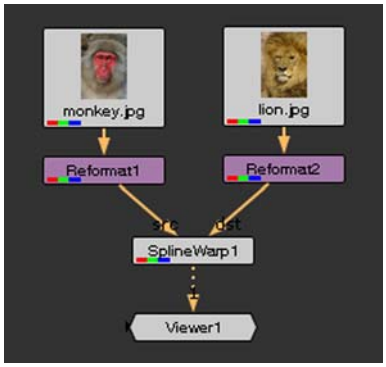
10. Attach the Viewer to the destination image or its Reformat node if one exists. You should now see the destination image and the blue destination grid.
11. Adjust the destination grid to conform to the subject of the destination image. This way, you warp the source image into the shape of the destination image.
12. Attach the Viewer to the GridWarp node and scrub to the frame where you want the morph to begin.
13. In the GridWarp controls, bring the **distortion** slider down to 0 (the source image). Click on the animation button and select **Set key**. Then, scrub to the frame where you want the morph to end and set **distortion** to 1 (the destination image).

14. Animate the **blend** control in the same way. Scrub to the frame where the morph starts, and set **blend** to 0 (the source image). Click the animation button and select **Set key**. Scrub to the frame where the morph ends, and enter 1 (the destination image) as the **blend** value.

If you now play the sequence in the Viewer, you'll notice that the source image is morphed into the destination image.

To morph one image into another using the SplineWarp node:

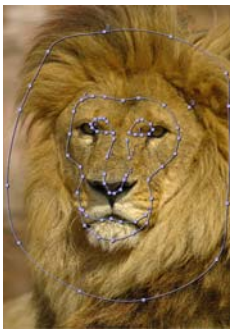
1. Select **Image** > **Read** to import the two images you want to morph together.
2. If the images do not have the same resolution, insert Reformat nodes after them. For more information, see *Reformatting Image Sequences* on page 99.
3. Select **Transform** > **SplineWarp** to insert a SplineWarp node into your script.
4. Connect the source image (the image you want to turn into another image) to the **src** input of the SplineWarp node, and the destination image (the image you want the source image turned into) to the **dst** input. Attach a Viewer to the SplineWarp node.



5. In the SplineWarp node's controls, set **show** to **src**. You should see the source image in the Viewer.

6. Identify some features that are similar in the source and the destination images. For example, if you are morphing together images of people or animals, these features might include their eyes, noses and mouths as well as the outlines of their faces and heads.
7. In the Viewer, **Ctrl**+**Alt**+**click** (Mac users **Cmd**+**Alt**+**click**) to draw curves around the key features you identified in the previous step. For more information on how to create the curves, see *To warp an image using the SplineWarp node:* on page 252.

8. Click **copy** under **Source Curves** and **paste** under **Destination Curves**. This copies the source curves you created in the previous step into the destination curves, making them identical.
9. From the **show** pulldown menu, select **dst**. You should now see the destination curves and the image connected to the **dst** input.
10. Adjust the destination curves to conform to the features of the destination image.



11. In the SplineWarp node's controls, select **blend** under **show**.
12. Scrub to the frame where you want the morph to begin. Bring the **distortion** slider down to 0 (the source image). Click on the animation button and select **Set key**. Then, scrub to the frame where you want the morph to end and set **distortion** to 1 (the destination image).

13. Animate the **blend** control in the same way. Scrub to the frame where the morph starts, and set **blend** to 0 (the source image). Click the animation button and select **Set key**. Scrub to the frame where the morph ends, and enter 1 (the destination image) as the **blend** value.

If you now play the sequence in the Viewer, you'll notice that the source image is morphed into the destination image.

11 Creating Effects

Several nodes in Nuke let you create various effects on your input images. In this chapter, we describe three of these nodes: LightWrap, Glint, and Text. The LightWrap node lets you create background reflections on foreground elements. The Glint node can be used to create star filter effects on image highlights. The Text node is useful for adding text overlays on images, for example, when creating slates or scrolling credits.

Background Reflections on Foreground Elements

You can use the LightWrap node to create background reflections on foreground elements. The node creates a reflection of light around the edges of your foreground element by blending in whatever is in the background.

The composite without the LightWrap effect

The composite with the LightWrap effect

Figure 11-1: Using the LightWrap node to create background reflections on the edges of the foreground element.

You may have noticed that objects filmed in front of very bright backgrounds have the appearance of softened edges as the light spills round from the background. When adding foreground layers onto a background plate, using the LightWrap node can dramatically improve the quality of your composite.

The composite without the LightWrap effect. Note the hard edges around the trailing hand and the bottom of the skirt.

The composite with the LightWrap effect. Note how the foreground fits better into the background.

Figure 11-2: The LightWrap node can improve composites with very bright backgrounds.

If you want to use LightWrap, you should apply it on your foreground element before you composite the foreground over the background with the Merge node.

To use the LightWrap node:

1. Select **Draw** > **LightWrap** to add a LightWrap node after your foreground and background images.
2. Connect your foreground element to input **A** of the LightWrap node, and the background image to input **B**.

3. Connect a Viewer to the output of the LightWrap node so you can see the effect of your changes.
4. Adjust the **Diffuse** and **Intensity** sliders to control both the spread and brightness of the reflections on the foreground element. These sliders need to be balanced out together. You may want to start by bringing Diffuse all the way down to better see what you are blending in from the background. Then, adjust Intensity before going back to the Diffuse slider and, if necessary, Intensity again until you are happy with the result.

low diffuse

high diffuse

low intensity

high intensity

5. If you want to create a uniform effect around the edges of the foreground rather than have the effect adjust itself according to the background, check **Disable luminance based wrap** on the **LightWrap** tab.
6. In case you don't want to merge the LightWrap effect with the foreground element in order to keep the LightWrap effect as a separate element, check **Generate wrap only** on the **LightWrap** tab.

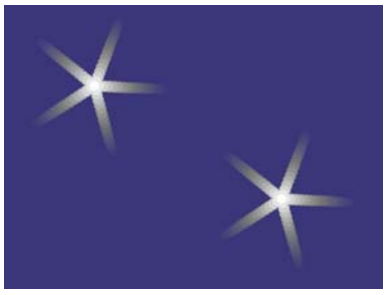
7. By default, the LightWrap effect is only applied inside the foreground element's alpha. If you want to extend the effect outside it, making the element seem to glow, check **Enable Glow**.

8. On the **Tweaks** tab, you can also adjust the following controls:
   - **FGBlur** to determine how much the foreground matte is blurred. The more blur, the more of the background is added to the foreground.
   - **BGBlur** to control how much the background is blurred before it is merged with the foreground element.
   - **Saturation** to adjust the saturation of the effect.
   - **Luma Tolerance** to increase or decrease the luminance values of the effect.
   - **Highlight Merge** to control how the foreground element is merged with the background. The default merge operation, called **plus**, adds the elements together, producing a glow effect.
   - Check **Use constant highlight** to use a constant colour of your choice rather than the background in the LightWrap effect. Select the colour using the controls next to **Constant Highlights Color**.
9. On the **CCorrect** tab, you can colour correct the LightWrap effect produced.

Creating Star Filter Effects on Image Highlights

With the Glint node, you can create star-shaped rays around all the bright points in an image.

The original image with bright points.



The image after using the Glint node.

To use the Glint node:

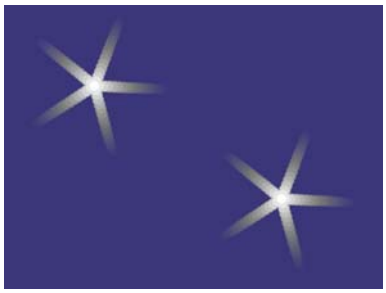
1. Select **Draw** > **Glint** to add a Glint node after the image you want to add star-shaped rays to.
2. From the **channels** pulldown menu and checkboxes, select the channels to which you want to apply the effect.
3. In the **no. of rays** field, enter the number of rays you want coming out of the bright points in your image. For example, if you want to create five-pointed stars, enter 5.
4. To change the threshold for how bright the highlights in the input image need to be to cause the glint effect, adjust the **tolerance** slider. Only the pixels above the threshold will bloom with the effect.

Low tolerance value.

High tolerance value.

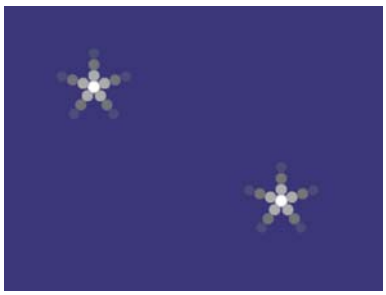
5. To determine the length of the rays, adjust the **length** slider. To give every other ray a different length and determine that length, adjust the **odd ray length** slider.
6. To determine how many steps the rays are formed of, enter a value in the **steps** field. The more steps you use and the shorter the rays are, the smoother the rays become.

The steps value set to 3.

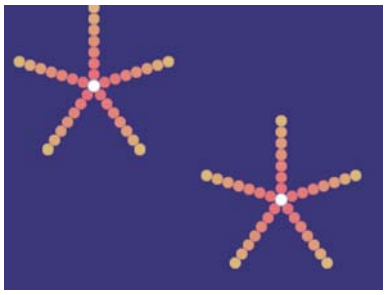


The steps value set to 5.

7. To rotate the star-shapes, adjust the **rotation** slider. Increasing the value rotates the rays clockwise, whereas decreasing the value rotates them counter-clockwise.
8. To change the colour in the beginning of the rays near the centre point of the stars, adjust the **from color** slider. To change the colour in the end of the rays, adjust **to color**. By default the from color is set to white, and the to color to black.



'From color' set to white, and 'to color' to black.



'From color' set to pink, and 'to color' to yellow.

9. If needed, you can also make the following adjustments:
   - If you want to change the aspect ratio of the stars, adjust the **aspect ratio** slider.
   - By default, the brightest image on the rays is used as the centre point for the star. However, if you prefer the images forming the rays to be added up in forming the centre point, uncheck **max**.

- To only output the Glint effect without merging it into the original input image used to create it, check **effect only**.

Glint effect on an input image.

Glint effect without the input image.

- To mask the shape that is used to create the rays, check **w** and select the mask channel from the pulldown menu.
- To perform a gamma correction on the highlights that cause glint before the glint effect is applied, adjust the **gamma** slider.
- To mask the glint effect, check **mask** and select a mask channel using the controls on the right.
- To dissolve between the original input image and the full glint effect, adjust the **mix** slider.

Creating Text Overlays

Using Nuke's Text node, you can add text overlays on your images. You can simply type in the text you want to have displayed or use TCL expressions or TCL variables to create a text overlay. Text overlays can also be animated so that their properties, such as position, size, and colour, change over time. These features make the Text node useful, for example, for creating slates or scrolling credits.

Creating a Text Overlay

To create a text overlay:

1. Select **Draw** > **Text** to create a Text node and connect a Viewer to it.
2. In the Text node properties, select the channels you want the text to appear in from the **output** controls.
3. If you want to multiply any channels by the drawn text so that they are set to black outside the text shape, select those channels using the **premult** controls.
4. From the **clip to** menu, select how you want to restrict the output image.
   - **no clip** - Do not restrict the output image.
   - **bbox** - Restrict the output image to the incoming bounding box.
   - **format** - Restrict the output image to the incoming format area.
   - **union bbox+format** - Restrict the output image to the combination of the incoming bounding box and format area.

- **intersect bbpx+format** - Restrict the output image to the intersection of the incoming bounding box and format area.

5. If you want to clear the affected channels to black before drawing on them, check **replace**. By default, **replace** is not checked and the text is drawn on top of the input image.
6. In the **message** field, enter the text you want to display, a TCL expression, a TCL variable, or a combination of these. For examples, see *Examples* on page 266.

   You should enter TCL expressions in square brackets, for example, **[date]**.

   To begin a new line, press **Return**.

   To display special Unicode characters, such as foreign language characters and copyright signs, you can

   - use decimal entities, such as **<** to display **<**, or **©** to display **©**
   - type Unicode characters, such as **<** or **©**, on your keyboard or cut and paste them from other applications. UTF-8 character encoding is used to store them in the control's value and in the saved Nuke script.

   The decimal entities only work if the font you are using has the character you want to display in it, and you have checked **turn "&#nnn;" into Unicode**.

Note

We recommend using decimal entities rather than typing **<**, for example. This is because future versions of the Text node may interpret HTML mark-up. In HTML, some characters, such as the greater than and less than signs, are reserved. If you used these signs within your text now, future versions could mistake them for HTML mark-up.

7. If you want to mask the effect of the Text operation, select the mask channel from the **mask** menu. To invert the mask, check **invert**. To blur the edges of the mask, check **fringe**.
8. To adjust the opacity of the text, use the **opacity** slider. The possible values run from 0 (invisible) to 1 (fully opaque).
9. To invert the inside and outside of the text shape, check **invert**.

Examples

The following table gives examples of TCL expressions, TCL variables, and decimal entities you can use in the **message** field of the Text node.

Message	Prints
TCL expressions	
[date]	Week day, day, month, hh:mm:ss, and time zone. For example, **Thu Jan 15 14:22:20 GMT**.
[date %a]	Abbreviated week day name. For example, **Thu**.
[date %A]	Full week day name. For example, **Thursday**.
[date %b]	Abbreviated month name. For example, **Jan**.

Message	Prints
[date %B]	Full month name. For example, **January**.
[date %d]	Day (01-31).
[date %D]	Date (dd/mm/yy). For example, **15/01/09**.
[date %H]	Hour (00-23).
[date %I]	Hour (01-12).
[date %m]	Month (01-12).
[date %M]	Minutes (00-59).
[date %p]	AM or PM.
[date %r]	Time (12-hour clock). For example, **11:04:07 AM**.
[date %S]	Seconds (00-59).
[date %T]	Time (24-hour clock). For example, **14:06:54**.
[date %y]	Abbreviated year (00-99). For example, **09**.
[date %Y]	Full year. For example, **2009**.
[date %z]	Numeric time zone. For example, **-0800**.
[date %Z]	Time zone. For example, **GMT**.
[frame]	Frame number. For example, **23**.
[value root.name]	Script directory path and script name. For example, **Users/john/Nuke_scripts/myscript.nk**.
TCL variables	
$env(*ENVIRONMENT_VARIABLE*)	The value of the environment variable specified. Replace *ENVIRONMENT_VARIABLE* with an environment variable you have set. For example, you can use **$env(USER)** to display the user name (for example, **john**) on Mac OS X and Linux, or **$env(USERNAME)** to display it on Windows and Linux. For a list of environment variables specific to Nuke, see *Nuke Environment Variables* on page 409.
$version_long	The full version number of Nuke. For example, 5.1v5.
$threads	Number of render threads used to calculate images. This is in addition to a main thread used to update the graphical user interface (GUI).
decimal entities	
£	£
©	©

Message	Prints
®	®
¿	¿
ê	ê
ß	ß
à	à

Tip

To get a list of all the TCL expressions you can use with **date**, type **x** on the Node Graph, set the script command dialog that opens to **TCL**, enter **date -h**, and click **OK**.

Repositioning and Transforming Text

Using the on-screen widget or the **Transform** controls on the Text node, you can apply geometric transformations (including translations, rotations, scales, and skews) to the text.

To transform text:

1. To reposition the text, drag the centre of the transformation overlay to a new location.
2. To translate, rotate, scale, or skew the text, use the transformation overlay. For more information on how to do this, see *Using the 2D Transformation Overlay* on page 159.

   Alternatively, you can adjust the **translate**, **rotate**, **scale**, and **skew** parameters in the Text controls.

Fonts

By using the FreeType library, the Text node supports a large number of fonts, including TrueType (.ttf) fonts and PostScript fonts (.pfa and .pfb).

To select a font:

1. In the Text controls, click on the folder icon next to **font**.

   The File Browser opens.
2. Browse to where you have stored the font and select it. To preview the font, click on the black arrow in the top right corner of the File Browser. Select **Open**.
3. Some font files contain more than one font face. If you are using such a font file, use the index field to the right of the **font** control to select the font face to use.

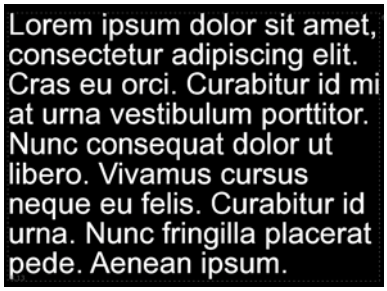
To adjust font size, spacing, and justification:

1. To adjust the size of the font, use the **size** slider. When **leading** is set to 0, this parameter also controls the spacing between each line of text.

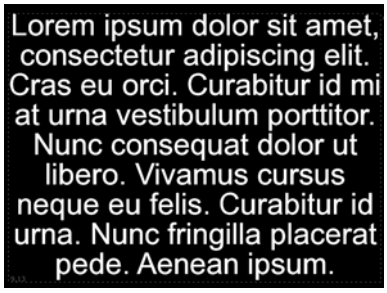
   When rendering the font, the **size** parameter controls the font hinting used. Font hinting adjusts which pixels are interpolated to more clearly render a font. At small sizes and on low resolution output devices, it has a big impact on the legibility of the font. For best

results, you should use the **size** parameter (rather than the **scale** parameter in the **Transform** parameter group) to control the size of the font and keep **scale** set to 1.

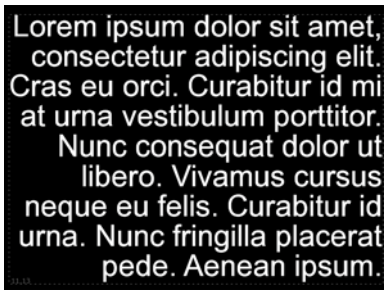
2. To increase or decrease the spacing between each letter, adjust the **kerning** slider. By using negative values, you can make the letters overlap.
3. If your text overlay contains several lines of text, you can adjust the spacing between each line by using the **leading** slider. By using negative values, you can make the letters overlap.
4. From the left-side **justify** menu, select how you want to align the text horizontally:
   - **left** - Align the text along the left edge of the on-screen text box. This leaves the right edge of the text ragged.


Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Cras eu orci. Curabitur id mi
at urna vestibulum porttitor.
Nunc consequat dolor ut
libero. Vivamus cursus
neque eu felis. Curabitur id
urna. Nunc fringilla placerat
pede. Aenean ipsum.

   - **center** - Align the text from the centre of the on-screen text box. This leaves both edges of the text ragged.

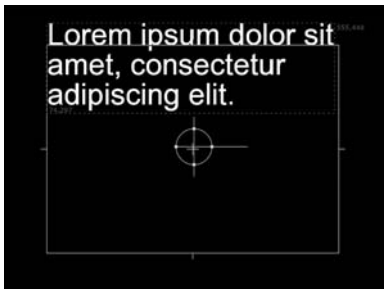

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Cras eu orci. Curabitur id mi
at urna vestibulum porttitor.
Nunc consequat dolor ut
libero. Vivamus cursus
neque eu felis. Curabitur id
urna. Nunc fringilla placerat
pede. Aenean ipsum.

   - **right** - Align the text along the right edge of the on-screen text box. This leaves the left edge of the text ragged.

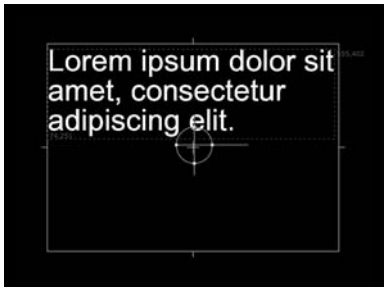

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Cras eu orci. Curabitur id mi
at urna vestibulum porttitor.
Nunc consequat dolor ut
libero. Vivamus cursus
neque eu felis. Curabitur id
urna. Nunc fringilla placerat
pede. Aenean ipsum.

5. From the right-side **justify** menu, select how you want to align the text vertically:

- **baseline** - Align the baseline of the first line of text along the top edge of the on-screen text box. The baseline is the imaginary line upon which most letters rest. This option allows you to reliably line up the baseline of different fonts.



- **top** - Align the text against the top edge of the on-screen text box.



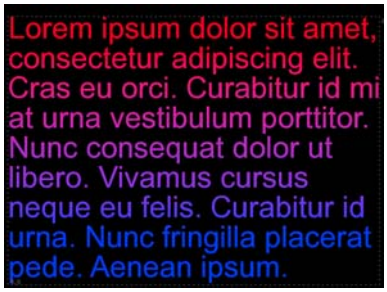
- **center** - Align the text from the centre of the on-screen text box.

- **bottom** - Align the text against the bottom edge of the on-screen text box.

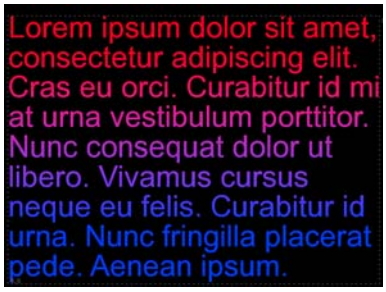
Changing the Text Colour

To change the text colour:

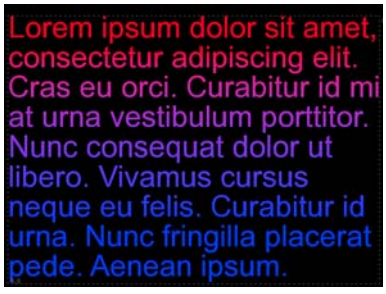
1. In the Text node controls, go to the **Color** tab.
2. Adjust the **color** parameter or click on the colour picker button to select a colour for the text.
3. If you want to create a colour gradient across the text, select anything other than **none** from the **ramp**:
   - **linear** - the ramp changes linearly from one colour into another.
   - **smooth0** - the ramp colour gradually eases into the point 0 end. This means colours in the point 0 end are spread wider than colours in the point 1 end.
   - **smooth1** - the ramp colour eases into the point 1 end. This means colours in the point 1 end are spread wider than colours in the point 0 end.
   - **smooth** - the ramp colour gradually eases into both ends. This means colours in the point 0 and point 1 ends are spread wider than colours in the centre of the ramp.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras eu orci. Curabitur id mi at urna vestibulum porttitor. Nunc consequat dolor ut libero. Vivamus cursus neque eu felis. Curabitur id urna. Nunc fringilla placerat pede. Aenean ipsum.

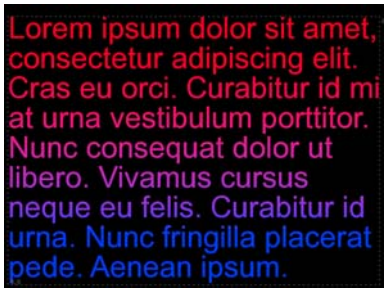
Linear ramp.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras eu orci. Curabitur id mi at urna vestibulum porttitor. Nunc consequat dolor ut libero. Vivamus cursus neque eu felis. Curabitur id urna. Nunc fringilla placerat pede. Aenean ipsum.

Smooth ramp.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras eu orci. Curabitur id mi at urna vestibulum porttitor. Nunc consequat dolor ut libero. Vivamus cursus neque eu felis. Curabitur id urna. Nunc fringilla placerat pede. Aenean ipsum.

Smooth0 ramp.


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras eu orci. Curabitur id mi at urna vestibulum porttitor. Nunc consequat dolor ut libero. Vivamus cursus neque eu felis. Curabitur id urna. Nunc fringilla placerat pede. Aenean ipsum.

Smooth1 ramp.

4. Use the **color** parameter to choose a colour for the ramp at the point 1 end (by default, the top end). Then, use **color 0** to choose the colour for the ramp at the point 0 end (by default, the bottom end).
5. To adjust the spread and angle of the ramp, drag the on-screen **point 0** and **point 1** controls to a new location. You may need to press **0** twice on the Viewer to make these controls appear.

Alternatively, you can enter new **x** and **y** values for the **point 1** and **point 0** parameters in the Text node properties.

12 Analysing Frame Sequences

This chapter concentrates on the CurveTool node. The node analyses an aspect of a frame sequence and creates an animation curve based on the analysis. You can then use the curve data to drive effects elsewhere. For instance, you can add matching flicker to a CG render.

Analysing and Matching Frame Sequences

You can use the CurveTool node to analyse four different aspects of your frame sequence, depending on which curve type you choose in the node controls:

- **AutoCrop** finds black regions (or any colour you pick) around the edges of the frame sequence and tracks their size and position over time. This is useful for running a Crop node to remove unnecessary outer pixels and speed up the calculation.
- **Avg Intensities** is useful for obtaining the average pixel values in a frame sequence and then matching that intensity elsewhere. It takes the first value in the frame range and the next value selected, adds them together and divides by two, returning the average between the two. You might want to use it to match the background plate's fire flicker in the smoke in the foreground plate, for example.
- **Exposure Difference** analyses the exposure changes in the frame sequence. It takes the first value in the frame range and the next value selected, and returns the difference between the two. You can use the results to match the same exposure elsewhere.
- **Max Luma Pixel** tracks the brightest and dimmest pixels in the frame sequence. This can be useful in the following case, for example. Let's say you have a night-time sequence depicting a person moving inside a dark house holding a flashlight, and want to add lens flare on the moving flashlight. Knowing where the brightest pixel is located over time allows you to match-move the lens flare and position it correctly without having to manually animate it.

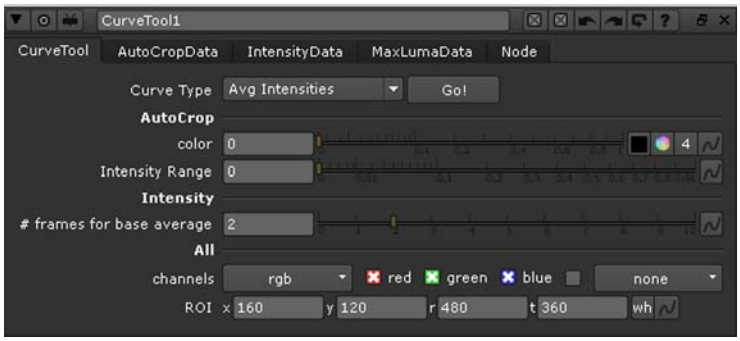


Figure 12-1: The CurveTool node properties panel.

Tip

If you are familiar with Shake, you may have used the PixelAnalyzer node. The CurveTool node is the Nuke equivalent of PixelAnalyzer.

To crop black edges to eliminate unnecessary computation:

1. Choose **Image** > **CurveTool** to insert a CurveTool node after the image sequence you want to analyse.
2. Make sure a Viewer is connected to the CurveTool node.

3. In the CurveTool controls, select **AutoCrop** from the **Curve Type** menu.
4. Using the **color** parameters, select the colour you want to track.
5. To control how far the colour can deviate from the selected colour and still be cropped off, use the **Intensity Range** slider.
6. From the **channels** menu and checkboxes, select the channels you want to analyse.
7. If you want to analyse an area in the frames rather than entire frames, define a region of interest either by dragging the edges of the frames to a new position in the Viewer, or by defining the area using parameters labeled **ROI**.
8. Click **Go!** to analyse the frames. This opens the *Frames to Execute* dialog.
9. In the dialog, define the frames to analyse. Enter the first frame, followed by a comma and the last frame. Click **OK**. Nuke starts analysing the frame sequence.
10. You'll find the results of the analysis on the **AutoCropData** tab where the parameter values have turned cyan to indicate they are animated over time. To see the animation curve, right-click on a parameter field and select **Curve editor**.

Once Nuke has created the animation curve, you can copy the animation or any of its values into a Crop node, for example, to match the analysed crop area there. **Ctrl**/**Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

To analyse the intensity of a frame sequence:

1. Select **Image** > **CurveTool** to add a CurveTool node in an appropriate place after the image sequence you want to analyse and match.
2. Connect a Viewer to the CurveTool.
3. In the node's controls, select **Avg Intensities** from the **Curve Type** pulldown menu.
4. Select the channels you want to analyse from the **channels** menu and checkboxes.
5. By default, the region of interest that is to be analysed covers the entire frame. If you want to analyse a smaller area, resize and reposition the region of interest in the Viewer by dragging its edges to a new position. You can also resize the region of interest using the **ROI** parameters in the control panel.
6. In the **# frames for base average field**, enter the range of frames that each frame being analysed is compared against. The frames are compared onwards from each frame analysed. Thus, a value of 1 would compare each frame to the frame following it, whereas a value of 5 would compare each frame to the following 5 frames.

   The higher frame range you use, the more accurate and time-consuming the calculation becomes. However, a high frame range is not always needed. For analysing and matching fire flicker, you'd probably want to go frame by frame, whereas removing flicker would require a wider frame range to ensure a good average is obtained as the result.
7. To analyse the sequence, click **Go!**. This opens the *Frames to execute* dialog.
8. In the dialog, specify the frame range you want to analyse and match. Enter the first frame, followed by a comma and the last frame. Click **OK**. Nuke now analyses the frame sequence.

9. Move to the **IntensityData** tab where you'll find the results of the analysis. You'll notice that the parameter input fields have turned cyan. This indicates that they are animated. To see the animation curve, right-click on the values and select **Curve editor**.

Once Nuke has created the animation curve, you can copy the animation or any of its values into a colour correction node, for example, to match the analysed intensity there. **Ctrl**/**Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

To analyse the exposure differences in a frame sequence:

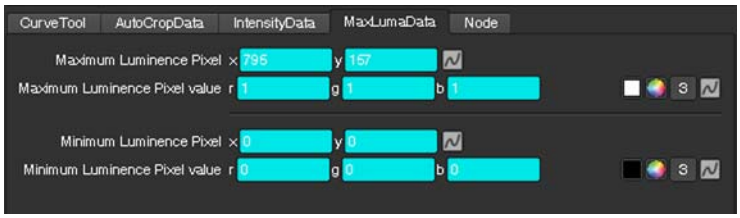
1. Select **Image** > **CurveTool** to add a CurveTool node after the image sequence you want to analyse.
2. Add a Viewer after the CurveTool.
3. Under **Curve Type**, select **Exposure Difference**.
4. From the **channels** pulldown menu and checkboxes, choose the channels you want to analyse.
5. If you want to analyse an area in the frame rather than the entire frame, define a region of interest either by dragging the edges of the frame box to a new position in the Viewer, or by defining the area using parameters labeled **ROI**.
6. To analyse the sequence, click **Go!**. The *Frames to Execute* dialog opens.
7. Specify the frame range you want to analyse. Enter the first frame, followed by a comma and the last frame. Click **OK**. Nuke now performs the analysis.
8. You can find the results of the analysis on the **IntensityData** tab where the parameter input fields have turned cyan to indicate they are animated. To see the animation curve, right-click on one of the input fields and select **Curve editor**.

Once Nuke has created the animation curve, you can copy the animation or any of its values into a colour correction node, for example, to match the analysed exposure there. **Ctrl**/**Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

To track the brightest and darkest pixels in a frame sequence:

1. Choose **Image** > **CurveTool** to add a CurveTool node after the image sequence you want to analyse.
2. Connect a Viewer to the CurveTool.
3. From the **Curve Type** menu, select **Max Luma Pixel**.
4. Click **Go!** to analyse the frame sequence. This opens the *Frames to Execute* dialog.

5. Define the frame range you want to analyse. Enter the first frame, followed by a comma and the last frame. Then, click **OK**. Nuke analyses the frame sequence, tracking both the position and the values of the brightest and darkest pixels.
6. You can find the results of the analysis on the **MaxLumaData** tab. You'll notice that the input fields have turned cyan to indicate that they are animated over time. To see the animation curve, right-click on an input field and **Curve editor**.



Once Nuke has created the animation curve, you can copy the animation or any of its values into another node to match that node's effect with the brightest or darkest pixels in the frame sequence. **Ctrl/Cmd**+click on the animation button and drag and drop it to another parameter to create an expression linking the two.

13 3D Compositing

Nuke's 3D workspace allows you to setup a 3D composite for camera moves, set replacement, and other applications where you need to simulate a "real' dimensional environment.

Although the 3D workspace has many potential uses, you're most likely to use it—at least initially—to create pan-and-tile scenes. These are scenes with 2D image planes arranged into a curved shape, and then rendered out through an animated camera to give the illusion of a seamless environment.

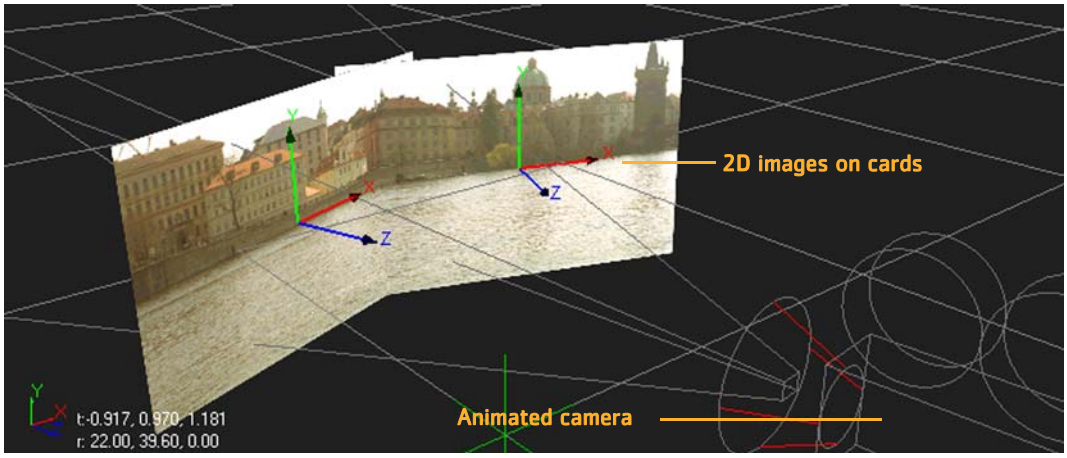


Figure 13-1: Simple pan-and-tile scene.

Overview

This chapter explains how to set up a 3D scene in Nuke, and how to add objects and cameras in the 3D workspace. You'll also see how to texture objects, transform objects and cameras, and render out scenes for use in other areas of your script.

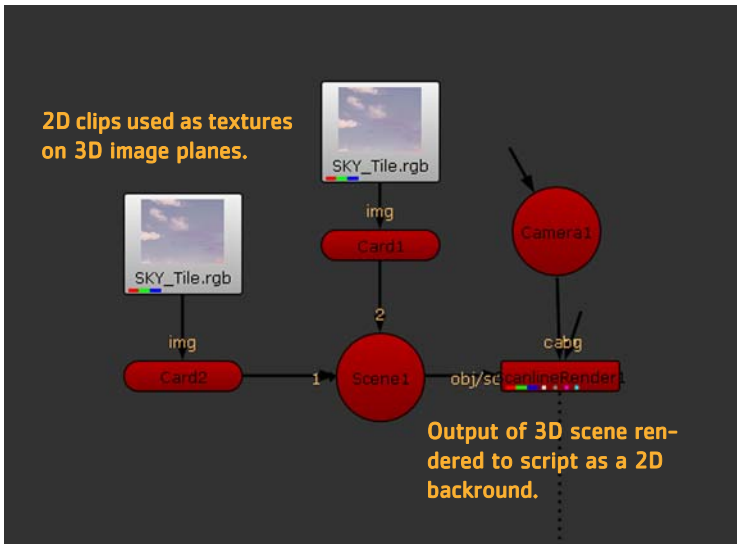


Figure 13-2: Script with 2D and 3D operators.

The 3D objects in Nuke appear as round shapes to differentiate them from objects that perform 2D operations. As shown above (Figure 13-2), you can mix 2D and 3D objects together in the node tree. For example, you can texture a 3D object with a 2D clip, or take the rendered output from a 3D scene and use it as a 2D background.

Setting Up a Scene

Each 3D scene includes the following objects: a Scene node, a Camera node, one or more geometry nodes (i.e., card, sphere, obj), and a ScanlineRender node. Examples of 3D scenes are shown in Figure 13-2 and Figure 13-3. In the example shown in Figure 13-2, the Scene node receives the output from two geometry nodes (*Card1* and *Card2*) and sends the composite of those objects to the ScanlineRender node, where the output is converted back to 2D.

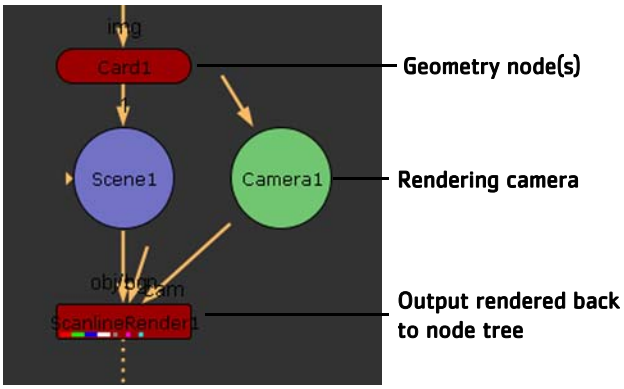


Figure 13-3: Core nodes for a 3D composite

Your script may contain multiple Scene nodes, cameras, and 3D render nodes. All 3D objects loaded in the Properties Bin will appear in the 3D Viewer, regardless of whether they are connected to the same Scene node.

The Scene Node

Regardless of its location in your script, the Scene node is the highest-level node in the scene hierarchy because it references all the elements in a 3D workspace—all the geometric objects, cameras, and materials.

To add a Scene node:

Choose **3D** > **Scene** from the toolbar.

The ScanlineRender Node

Every Scene node in a script should be connected to a ScanlineRender node, which tells Nuke to render the results of the scene. The ScanlineRender node also allows you to toggle between a 2D and 3D view of the scene.

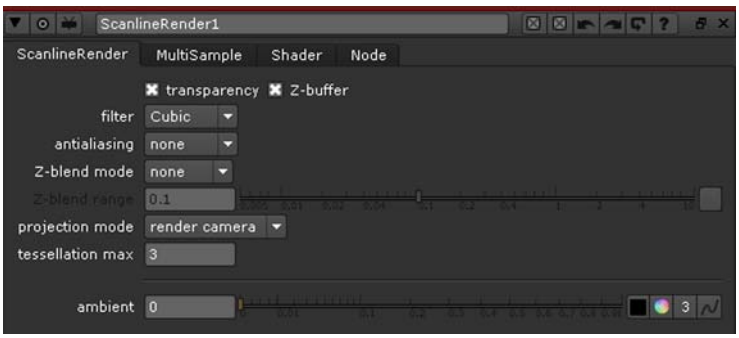


Figure 13-4: Parameters for the ScanlineRender node.

To add a ScanlineRender node:

1. Select the Scene node.
2. Choose **3D** > **Render** from the toolbar.
3. Connect the **obj/scn** input to a Scene or geometry node.
4. Connect the **cam** input to the main camera.
5. Connect the optional **bg** input to composite a background image into the scene.
6. Press **Ctrl+I** (**Cmd+I** on a Mac) to open a new Viewer to display the output of the ScanlineRender node.

When an image is connected to the bg input, its resolution becomes the output resolution for the ScanlineRender node.

The Camera Node

Cameras may be connected to either the Scene node or the ScanlineRender node. The camera connected to the ScanlineRender node is the camera used for rendering.

1. Choose **3D** > **Camera** to insert a camera node.
2. Drag an output connector from the Camera node to a Scene node or connect the Camera node to a ScanlineRender node's cam input.

When connecting cameras for the 3D scene, the camera you want to use for rendering should be connected to the ScanlineRender node, like this:

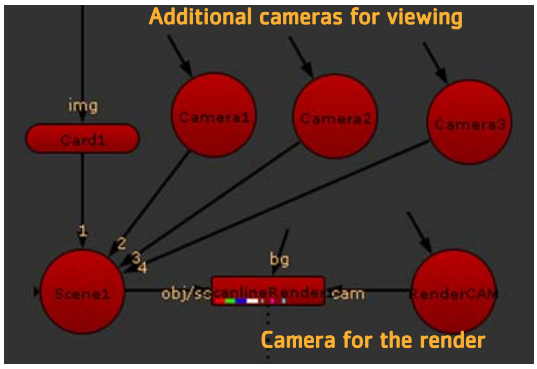


Figure 13-5: Connecting cameras to the scene.

Any additional cameras should be connected to the Scene node. When you have multiple cameras associated with a 3D scene, you can switch between them by choosing the viewing camera from the list at the bottom of the Viewer. See the next section, *Using the 3D Viewer*, for more information.

Using the 3D Viewer

When you have a 3D setup in your script, any Viewer window can toggle between the 2D and 3D display modes. The 2D mode shows the result of a rendered scene, the 3D mode shows the perspective from one camera in the scene.

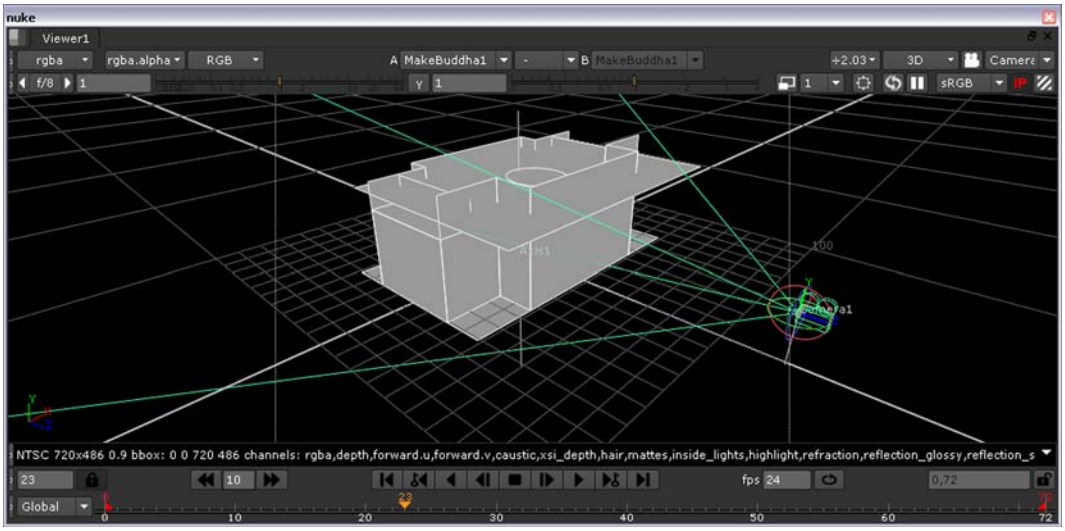


Figure 13-6: The 3D Viewer

When you do not have a Camera node in your script, the 3D Viewer uses default views (see Figure 13-7 below for the list of options). These views are similar to different cameras that you can look though, but they don't appear as objects that you can manipulate in the scene.

Switching to the 3D Viewer:

Open a Viewer and press **Tab** or **V** to toggle between the 2D and 3D modes— or select the view you want from the list at the top right corner of the Viewer window.

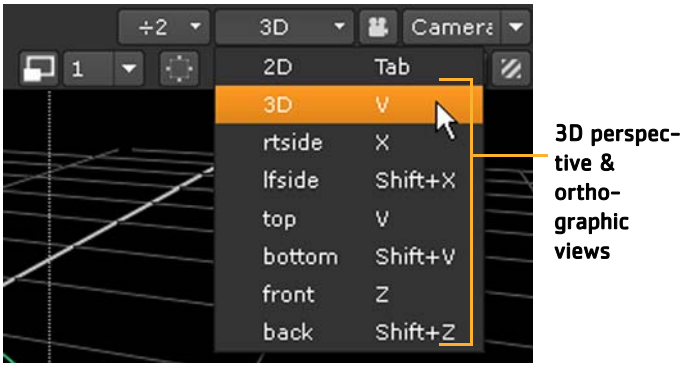


Figure 13-7: Switching to the 3D Viewer

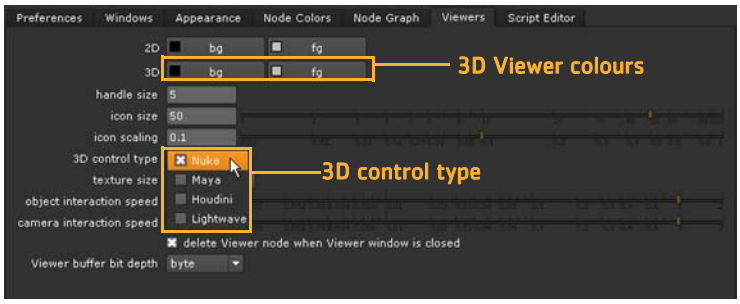
The "built-in" views give you different perspectives on your 3D scene. You can quickly switch between the views by pressing the hot keys for right view (**X**), left view (**Shift+X**), top view (**C**), bottom (**Shift+C**), front (**Z**), back (**Shift+Z**), and three-quarter perspective (**V**).

To navigate in the 3D Viewer:

- *Dolly*: Press **Alt** and **middle-mouse-button** drag.
- *Pan*: Press **Alt** and **left-mouse-button** drag.
- *Tilt*: Press **Ctrl/Cmd** and **left-mouse-button** drag.
- *Spin*: Press **Ctrl/Cmd** and **left-mouse-button** drag.
- *Roll*: Press **Ctrl/Cmd+Shift** and **left-mouse-button** drag.
- *Look through camera*: Select a camera object, press **H**.
- *Fit the scene*: Press **F** to fit the entire 3D scene within the Viewer.

To change the 3D Viewer display properties:

1. Open the Preferences window (**Shift+S**), and select the **Viewers** tab.



2. Make the desired changes to the 3D **bg** and **fg** colours.
3. From the **3D control type** list, select the navigation control scheme you want to use (Nuke, Maya, Houdini, or Lightwave).
4. Click **Save Prefs**.

Important

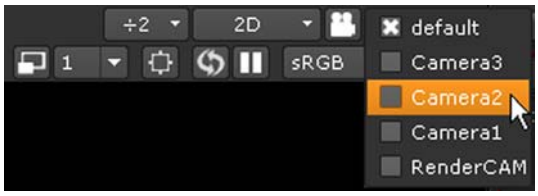
The 3D control type also affects the mouse button assignments for panning and zooming in the node graph and 2D Viewers.

To look through a camera:

1. Press **V** to make sure you are looking through the 3D perspective view, and not one of the orthographic views.
2. Select the camera in the Viewer or select the camera's node in the workspace. Then press **H** (home).

   or

   From the 3D Viewer window, select the camera from the list in the top right corner.



Note

This selection does not change the camera used for rendering. This changes only the camera to "look through" for the current 3D Viewer.

Cameras in the current data stream automatically appear in the list of cameras you can select. To select a camera that doesn't appear in list, double-click the camera node to open its panel, and it will be added to the list.

3D Scene Geometry

Nuke includes several options for inserting 3D geometry into your scenes. You can create primitive shapes, such as cards, cubes, and spheres, as well as import models created in other 3D applications.

These are the types of objects you can include in a Nuke 3D scene, and each objects is represented by a 3D node in the script:

- Cards
- Cubes
- Cylinders
- Spheres
- OBJ (Wavefront) objects
- Axes
- Cameras
- Lights

Working with Cards

A *card* is the simplest type of object you can add to a scene (and probably the type you will use most often). It's merely a plane onto which you can map a texture—typically a clip you are using as part of a pan-and tile setup.

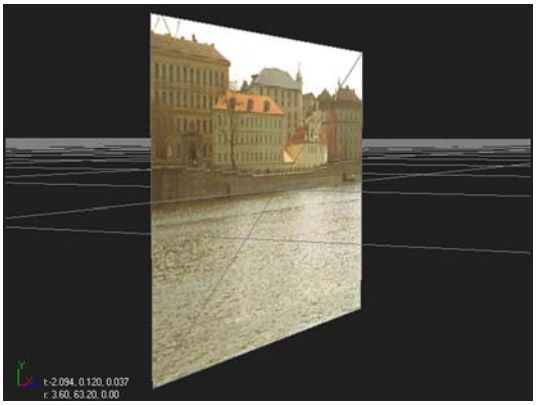


Figure 13-8: A card object

A card object may be deformed as a bilinear or bicubic object with controls contained in the card's parameters. You can also insert other 3D nodes, such as ProceduralNoise or RadialDistort, to change the card geometry.

Card nodes have extended bicubics (bicubics with more control points). They allow you to subdivide a card, giving you finer control for warping an area. You can subdivide the card into an evenly spaced grid or pick a location to add a row, column, or both.

To add a card object:

1. Click **3D** > **Geometry** > **Card** to insert a Card node.
2. Drag the Card node's **img** pipe to the Read node that has the image you want to apply to the card.
3. Connect the Card node to the appropriate Scene node to add it to the 3D scene.
4. Use the card object's transform controls to manipulate the position, scale, and rotation of the card in 3D space. For more information, see *Transforming from the Node Properties Panel* on page 299.

Deforming Card Objects

The Deform tab on the Card panel lets you convert the card into a mesh surface that may be pulled and reshaped.

A *bicubic deformation* offers the greatest degree of surface elasticity. You can add any number of control points on the card and translate these points and their tangents in any direction. The control point tangents exert a magnetic-like influence over the objects surface.

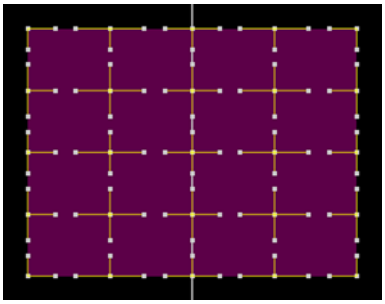
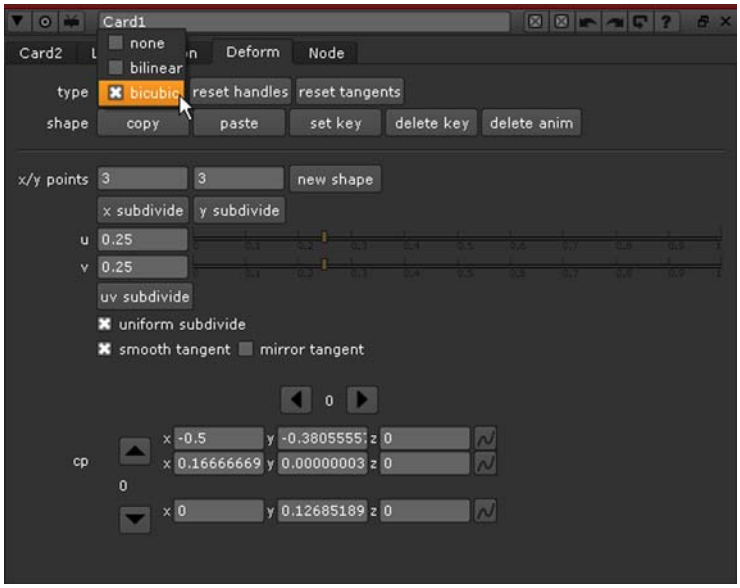


Figure 13-9: The Card node can have any number of control points you can translate.

To deform a Card object:

1. Double-click the Card node to open its controls.
2. Go to the **Deform** tab, and select the mesh type for the deformation: **bilinear** or **bicubic**.



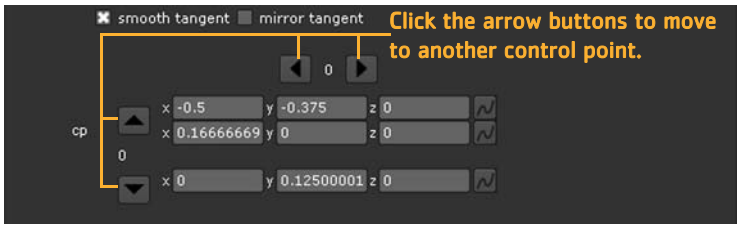
3. By default, the card has three control points on the x axis, and three on the y axis. To add more control points, do any of the following:
   - Enter new values in the **x/y points** fields and click the **new shape** button. For example, to create a shape with 4 points on the x axis and 6 on the y axis, change the **x points** value to 4 and the **y points** value to 6, and click **new shape**.
   - To evenly subdivide the current shape in the x or y directions, click the **x subdivide** or **y subdivide** buttons. This adds one control point between every existing control point in the chosen direction. The **x/y points** fields are also updated to reflect the current number of control points.
   - To add one row or column of control points, adjust the **u** or **v** slider. The **u** slider specifies the position of new columns, and the **v** slider the position of rows. In the Viewer, a yellow cross marker indicates the position of the new row or column. You can also move the cross marker by dragging it to a new position in the Viewer. The **u** and **v** sliders' values are updated as you move the marker. When you are happy with the

position, click the **uv subdivide** button. A row or column is added in the position you specified. Clicking the button again has no effect, because there is already a subdivision at the specified position.

4. If you selected **bicubic** under **type**, you can adjust the way control point tangents behave when you are making your changes to the card. Do any of the following:
   - To have the original tangents adjusted to create a more uniform subdivision when you are using **x subdivide**, **y subdivide**, or **uv subdivide**, check **uniform subdivision**. If you do not check this, Nuke maintains the original tangents.
   - You can move the tangents in the Viewer by clicking and dragging. If you want to move a tangent together with the opposite tangent so that the two tangents form a continuous line, check **smooth tangent**. To break the tangent from the opposite tangent and move the tangent alone, uncheck **smooth tangent**.
   - To change the length of the opposite tangent to always match the length of the tangent you are moving, check **mirror tangent**. If you do not check this, the opposite tangent length is not changed.
5. Drag the points displayed in the mesh to deform the card.

To translate the control points and tangents:

1. If necessary, double-click on the Card node to display its controls, and go to the **Deform** tab.
2. Only the controls for the selected point are displayed in the bottom of the Card properties panel. To translate another point, you can select a new point in the Viewer or use the arrow buttons in the bottom of the Card controls to move to the next control point.



3. To translate the control points and their tangents:
   - Increment or decrement the numbered **x**, **y**, and **z** fields. For each control point, the controls for translating the point itself are shown on top of the controls for translating the tangents.
   - Or drag on any control point or tangent to translate it relative to the current angle of view.

Working with Cubes

A *cube* is the familiar six-sided polyhedron. You can transform any of its sides (and, of course, texture it with clips).

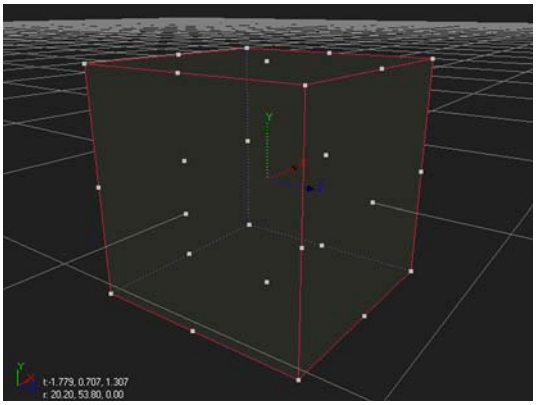


Figure 13-10: A cube object

To add a cube:

1. Click **3D** > **Geometry** > **Cube** to insert a Cube node.
2. Drag the Cube node's **img** pipe to the Read node containing the clip you want to use as a texture.
3. Drag one of the Scene node's numbered pipes to the Cube node to place the cube in the scene.
4. Use the cube object's transform controls to manipulate the position, scale, and rotation of the cube in 3D space. For more information, see *Transforming from the Node Properties Panel* on page 299.
5. Translate any of the cube's sides to alter its shape.

To translate a cube's sides:

1. If necessary, double click on the Cube node to display its parameters (and thereby select the object in the scene).
2. Increment or decrement the **cube** fields. (Assuming a positive z view of the object, **x** refers to the left side; **y**, the bottom side; **n**, the back side; **r**, the right side; **t**, the top side; and **f**, the front side.)

   Or drag on any side order to translate it relative to the current angle of view.

Working with Spheres

A *sphere* is the familiar globe-shaped polyhedron. You can control its geometric resolution, or face count (and, of course, texture it with clips).

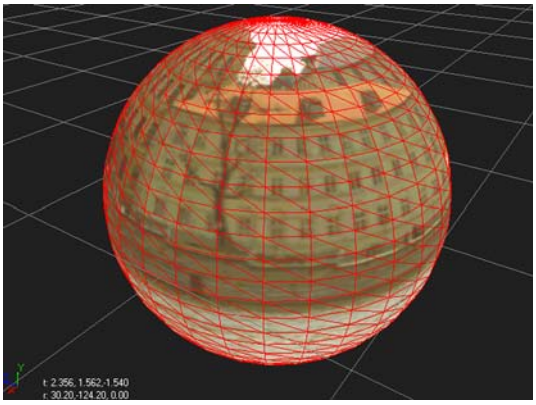


Figure 13-11: A sphere object

To add a sphere:

1. Click **3D** > **Geometry** > **Sphere** to insert a Sphere node.
2. Drag the Sphere node's **img** pipe to Read node containing the clip you want to use as a texture.
3. Drag one of the Scene node's numbered pipes to the Sphere node to place the bicubic object in the scene.
4. Use the sphere object's transform controls to manipulate the position, scale, and rotation of the sphere in 3D space. For more information, see *Transforming from the Node Properties Panel* on page 299.

Adjusting Geometric Resolution

By default, a sphere has 30 rows and 30 columns. You can, however, increase or decrease either number as appropriate. For example, the figure below shows a sphere whose geometric resolution has been decreased to 2 rows and 4 columns (making it, in effect, an octahedron).

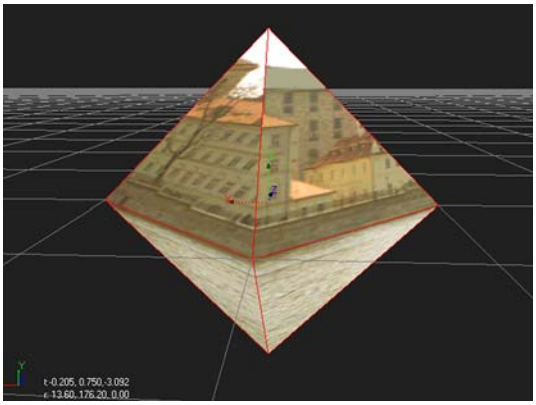


Figure 13-12: An octahedron generated with a low-resolution sphere

To adjust a sphere's geometric resolution:

1. If necessary, double click on the Sphere node to display its parameters (and thereby select the object in the scene).
2. Increment or decrement **rows** field to adjust the number of latitudinal divisions on the sphere.
3. Increment or decrement **columns** field to adjust the number of longitudinal divisions on the sphere.

Working with OBJ Objects

You can import into a Nuke scene 3D objects from other software programs which have been saved out in the OBJ (Wavefront) format. You cannot manipulate OBJ objects at the vertex level from inside Nuke, but you can texture and transform them.

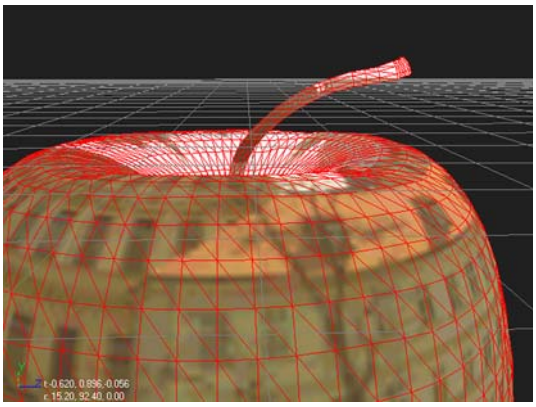


Figure 13-13: An imported OBJ object

To import an OBJ object:

1. Click **3D** > **Geometry** > **ReadGeo** to insert a ReadGeo node.
2. In the ReadGeo parameters, click the **file** field's folder icon. The file navigation dialog appears.
3. Navigate to the OBJ file, then click **Open**. Nuke reads in the OBJ file.
4. Drag the ReadGeo node's **img** pipe to the Read node containing the clip you want to use as a texture.
5. Drag one of the Scene node's numbered pipes to the ReadGeo node to place the OBJ object in the scene.

Parenting to Axis Objects

An axis object works as a null object by adding a new transformational axis to which other objects may be parented. Even when objects already have their own internal axes, it's sometimes useful to parent in a separate axis.

For example, the Axis node has been parented to the other objects in the scenes (the two image planes and the camera). The result is an axis object which globally controls the scene. Rotating it, for example, rotates all objects in the scene, as the figure below shows.

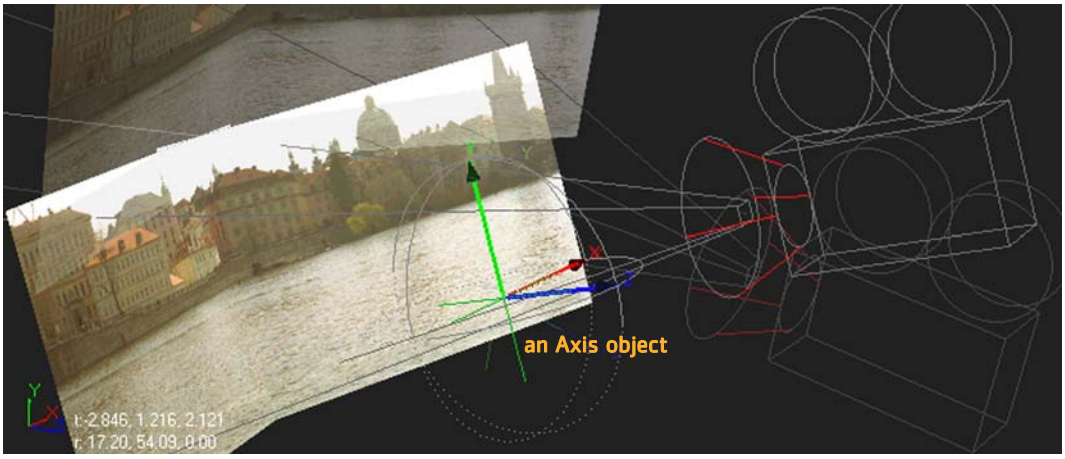


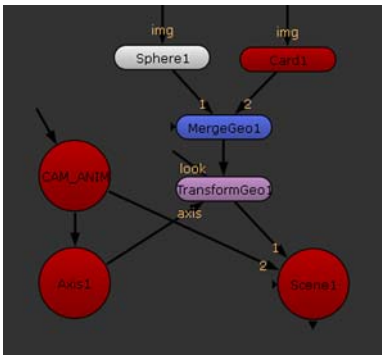
Figure 13-14: Rotating an entire scene with an axis object

Tip

To move several objects together, you can also merge them using a MergeGeo node (see *Merging Objects* on page 290) and then control them using a TransformGeo node (see *Using the TransformGeo Node* on page 300).

To add an axis object:

1. Click **3D** > **Axis** to insert an Axis node.
2. To create the parent relationships, connect the Axis node all object nodes you wish to control with the Axis transformation controls.



To create a nested transformational hierarchy, chain additional Axis nodes to the first one you inserted. For example, you could create a hierarchy of three Axis nodes to control rotation, scale, and transform.

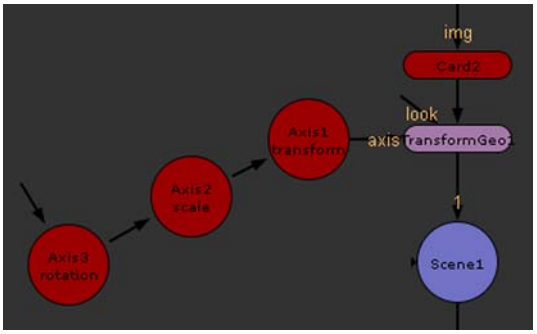


Figure 13-15: Creating a nested transformational hierarchy

In the above example, *Axis3* (rotation) is connected to *Axis2* (scale). *Axis2* is connected to *Axis1* (transform), and *Axis1* is connected to the TransformGeo node(s) that you want to affect. With the Axis nodes connected in this manner, their transformation data ripples down the chain and is added to the settings of the TransformGeo node.

Merging Objects

With the MergeGeo node, you can merge your 3D objects together to process all of them at the same time. For example, after merging your objects, you can use a Transform node to move the objects together, or add an ApplyMaterial node to apply a global material to them.

To merge your 3D objects:

1. Select **3D** > **Modify** > **MergeGeo** to insert a MergeGeo after the 3D objects in your script.
2. Connect the objects you want to merge to the MergeGeo node's inputs.

   You can now process all the objects you connected to the MergeGeo node together.

Object Material Properties

The nodes under the Shader menu let you define the material attributes of geometric objects in a scene, including the quality of light reflected back to the camera from an object's surface. Using these nodes, you can control what material your objects seem to be made of.

You can also add several Shader nodes one after the other to produce more complex effects. For this, you should use the unlabeled inputs on the Shader nodes.

The material property settings you apply affect the render output of the scene.

You can insert 3D shader nodes in the following places in your scripts:

- between the 2D image you're using for the surface texture and the 3D object node that creates the surface.

- after the 3D object nodes using the ApplyMaterial node. Connect the geometry (for example, a Sphere node) to the ApplyMaterial node, and the materials (for example, a BasicMaterial node) to the ApplyMaterial node's mat input. This is a good way to apply a global material to all objects.

You can use the **map** connectors to input a mask image to limit the effect of the material change.

Note

To see the effect of changes to an object's material properties, transparency and lighting must be enabled for OpenGL rendering. Press **S** over the 3D Viewer, and check **transparency** and **headlamp** on the **3D** tab.

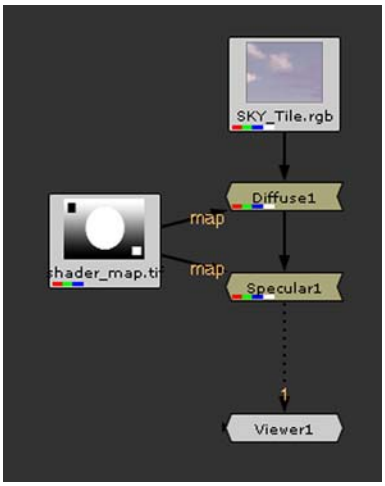
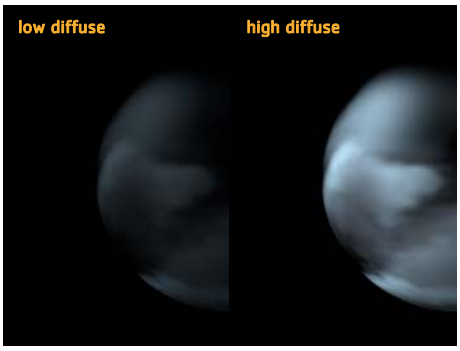


Figure 13-16: Diffuse and Specular nodes

To define material properties:

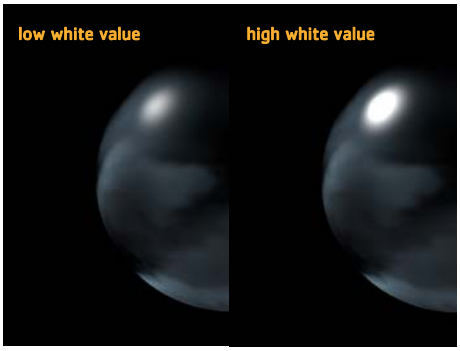
- Choose **3D** > **Shader** > **Diffuse** to insert a Diffuse node. This node lets you adjust the colour of the material when illuminated. The material appears darker as the surface points away from the light, as the light is not falling on it.

  Adjust the **white** slider in the Diffuse panel to control the diffuse colour. By default, this is in greyscale, but you can adjust the individual r, g, and b values. The higher the value, the brighter the material.

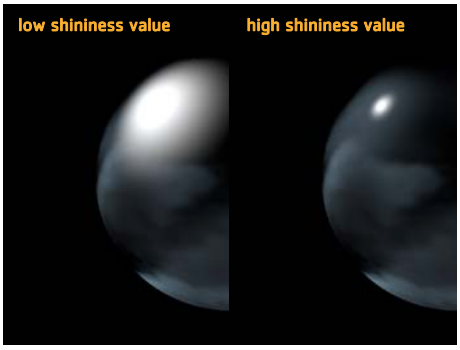


- Choose **3D** > **Shader** > **Specular** to insert a Specular node. You can use this node to control how bright and wide the highlights on the material seem. The location of the viewpoint is significant: the specular highlights are the brightest along the direct angle of reflection.

  Adjust the **white** slider to control the brightness of the specular highlight. The higher the value, the shinier the material seems.

  To control the width of the highlights, adjust the **min shininess** and **max shininess** sliders.

  Adjust **shininess channel** to control how the input channels are used to map the black and white values to the minShininess and maxShininess parameters when a **mapSh** input is connected. Choose **red** to use the red channel for the mapping, **green** to use the green channel, **blue** to use the blue channel, **luminance** to use the luminance, or **average rgb** to use the average of the red, green, and blue channels.

- Choose **3D** > **Shader** > **Emission** to insert an Emission node. You can use this node to simulate lamps or other sources that emit light. Adjust the **emission** slider to change the

brightness of non-illuminated areas for the surface. The higher the value, the more light the material seems to emit and the brighter it appears.

- Choose **3D** > **Shader** > **Phong** to insert a Phong node. This nose uses the Phong algorithm to provide more accurate shading and highlights.

  The Phong node has several map inputs you can use to mask the effect of the node. You can use:

  - **mapD** to modulate the diffuse component,
  - **mapS** to modulate the specular component,
  - **mapE** to modulate the emission component, and
  - **mapSh** to modulate the shininess value.

  You can adjust the following sliders in the node's controls:

  - **color** to change the material colour.
  - **emission** to change the colour of the light the material emits.
  - **diffuse** to control the colour of the material when illuminated.
  - **specular** to control how bright the highlights on the material seem.
  - **shininess** to control how shiny the material appears.
  - **min shininess** and **max shininess** to set the minimum and maximum shininess values. If you haven't connected an image to the **mapSh** input of the node, the average of these values is used as the shininess value for the material.
  - **shininess channel** to control how the input channels are used to map the black and white values to the minShininess and maxShininess parameters when a **mapSh** input is connected. Choose **red** to use the red channel for the mapping, **green** to use the green channel, **blue** to use the blue channel, **luminance** to use the luminance, or **average rgb** to use the average of the red, green, and blue channels.

- Choose **3D** > **Shader** > **Basic Material** to insert a BasicMaterial node. This node is a combination of the Diffuse, Specular, and Emission nodes, allowing you to control all three aspects of the material with a single node.

  Like the Phong node, the BasicMaterial node has several map inputs you can use to mask the effect of the node. You can use:

  - **mapD** to modulate the diffuse component,
  - **mapS** to modulate the specular component,
  - **mapE** to modulate the emission component, and
  - **mapSh** to modulate the shininess value.

  In the node's controls, you can adjust the following parameters:

  - **emission** to change the colour of the light the material emits. Note that when you have an image connected to the unlabeled input of the BasicMaterial node and adjust this

value, you need to look at the rendered 2D image to see the effect of your changes. Changing the emission value does not have any effect in the 3D Viewer.

- **diffuse** to control the colour of the material when illuminated.
- **specular** to control how bright the highlights on the material seem.
- **min shininess** and **max shininess** to set the minimum and maximum shininess values. If you haven't connected an image to the **mapSh** input of the node, the average of these values is used as the shininess value for the material.
- **shininess channel** to control how the input channels are used to map the black and white values to the minShininess and maxShininess parameters when a **mapSh** input is connected. Choose **red** to use the red channel for the mapping, **green** to use the green channel, **blue** to use the blue channel, **luminance** to use the luminance, or **average rgb** to use the average of the red, green, and blue channels.

Projecting Textures onto Objects

You can use the UVProject and the Project3D nodes to project texture images onto your 3D objects. This way, you can add detail, surface texture, or colour to your geometry, making the geometry more realistic and interesting.

The UVProject node changes the uv values of the vertices whereas the Project3D node is a material shader.

Projecting Textures with the UVProject node

The UVProject node sets the uv coordinates for the object, allowing you to project a texture image onto the object. If the object already has uv coordinates, this node replaces them.

To use the UVProject node:

1. Select **3D** > **Modify** > **UVProject** to insert a UVProject node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it.
4. Connect an Axis or a Camera node to the UVProject node's axis/cam input. If you connect an Axis node, project the texture uv coordinates onto the object using the axis transform values (that is, translation, rotation, scale, etc.). If you connect a Camera node, do a similar projection as with the axis but also use the camera lens information, such as the aperture.
5. Adjust the following parameters:
   - From the **projection** pulldown menu, select the projection type. Usually, it's best to select a type that's close to the object's surface shape. For example, if your object is a sphere, like a football or a planet, select **spherical**.
   - From the **plane** pulldown menu, select the projection direction: **XY**, **YZ**, or **ZX** to project the texture image along the z, x, or y axis. This pulldown menu is only available if you selected **planar** as the projection type.

- To mirror the texture uv coordinates in the horizontal direction, check **invert u**. To mirror them in the vertical direction, check **invert v**.
- To scale (stretch or squash) the texture uv coordinates in the horizontal direction, adjust the **u scale** slider. To scale them in the vertical direction, adjust the **v scale** slider. The higher the value, the more the texture is stretched.
- To change the name of the attribute that's used as the vertex's uv coordinates to find the image pixel, enter a name in the **attrib name** field.

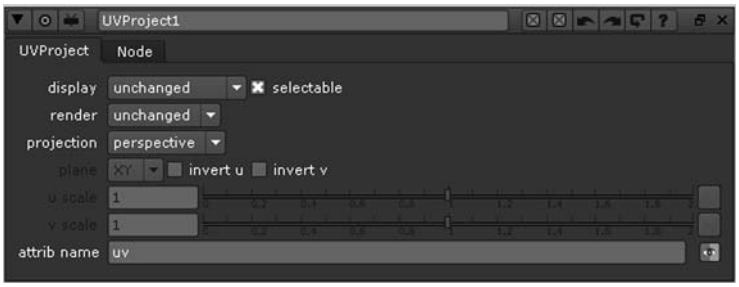


Figure 13-17: The UVProject node's properties panel.

Projecting Textures with the Project3D Node

The Project3D node projects an input image through a camera onto the 3D object.

To use the Project3D node:

1. Select **3D** > **Shader** > **Project3D** to insert a Project3D node after the image you want to project. Connect a Camera node to the Project3D node's cam input.
2. Insert a 3D geometry node (for example, a sphere) after the Project3D node.
3. Attach a Viewer to the 3D geometry node to see your changes.
4. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it.
5. From the **project on** pulldown menu, select to project the image on either the front facing, back facing, or both polygons.
6. To extend the input image at its edges with black, check **crop**. To extend the image with the edge colours, uncheck **crop**.

Merging Shaders

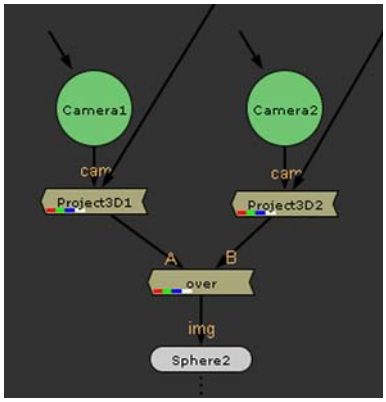
With the Shader menu's MergeMat node, you can combine two shader nodes together, using compositing algorithms like **none**, **replace**, **over**, and **stencil**. The MergeMat node is particularly useful for combining multiple Project3D nodes, allowing you to composite 2D images projected onto the 3D geometry atop each other.

You connect the shader nodes to the MergeMat node's A and B inputs. A refers to the foreground element, and B to the background element.

To merge shaders:

1. Select **3D** > **Shader** > **MergeMat** to add a MergeMat (over) node after the two shader nodes you want to combine.
2. Connect the MergeMat node to the **img** input of the 3D object you want to project the images on.

   For example, if you wanted to combine two Project3D nodes and composite their results onto a sphere, your node tree would look something like the following:



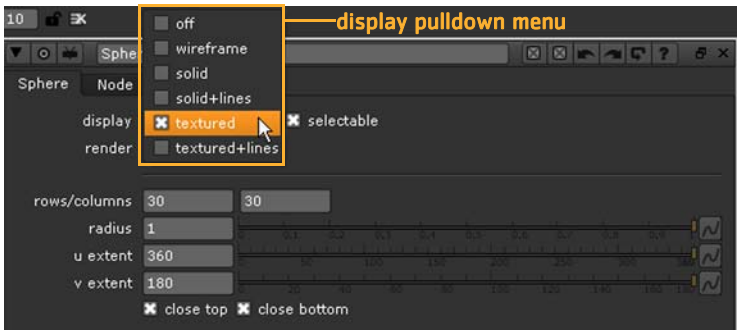
3. For operations (such as **over**) that need an alpha channel (mask), select which channel to use for the alpha from the **Alayer** pulldown menu.
4. From the **operation** pulldown menu, select how you want to composite the results of the two shader nodes together:
   - to only use input B in the composite, select **none**.
   - to only use input A in the composite, select **replace**.
   - to composite input A over input B using the mask, select **over**.
   - to use input B outside the mask area, select **stencil**.
   - to use input B inside the mask area, select **mask**.
   - to add input B to input A, select **plus**.
   - to use input A if it is greater than input B or else use input B, select **max**.
   - to use input A if it is less than input B or else use input B, select **min**.

Object Display Properties

You can adjust the display characteristics of all geometric objects in a scene. These settings don't affect the render output of the scene; these are for display purposes only in the 3D Viewer.

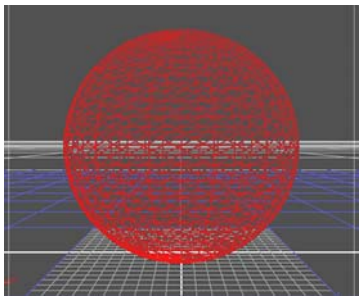
To edit an object's display attributes:

1. Double click on the object's node to display its parameters.
2. From the **display** list, choose the display type that you want for the object.

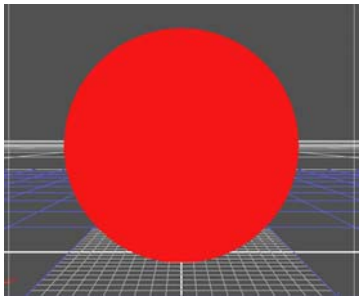


These are how each of the display options appear:

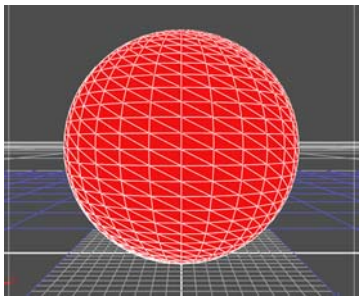
- **wireframe** displays only the outlines of the object's geometry.



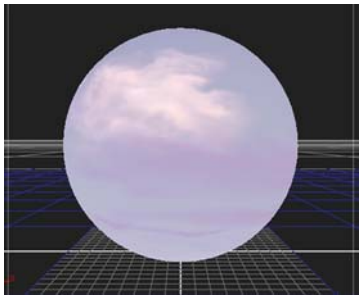
- **solid** displays all geometry with a solid colour.



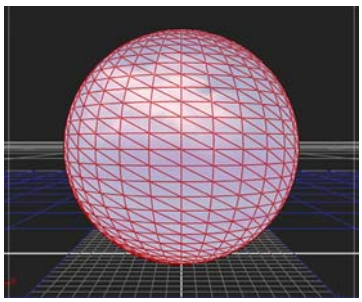
- **solid + lines** displays the geometry as solid colour with the object's geometry outlines.



- **textured** displays the only the surface texture.



- **textured + lines** displays the wireframe plus the surface texture.



Transforming Objects

Transform operations include moving, scaling, rotating the objects in your 3D scene. When an object node is active, you can enter specific transform settings in the node parameters, or directly manipulate the object with the transform handles displayed in the 3D Viewer. You can also link transform parameters to imported track or camera data, or control the transforms with animation curves.

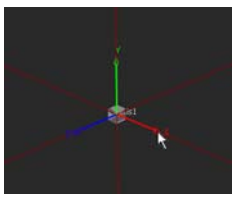
Cameras, geometry objects, and lights have their own set of transform controls built-in.

Using the Transform Handles

Transform handles appear when a 3D object with transform capabilities is loaded into the Properties Bin. The colours of the handles correspond to the axes available in 3D space: red transforms the x-axis, green transforms the y-axis, and blue transforms the z-axis.

To move an object with the transform handles:

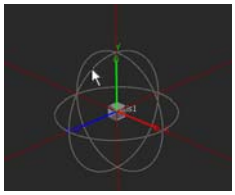
- Drag an object to move it on any axis.



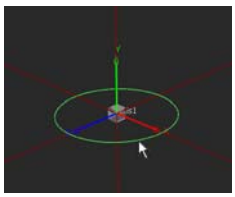
- **Shift**+drag to constrain movement to one axis.

To rotate an object with the transform handles:

- **Ctrl**+drag (Mac users **Cmd**+drag) to rotate the object on any axis.



- **Ctrl+Shift**+drag (Mac users **Ctrl**+**Shift**+drag) to constrain the rotation to one axis.



Transforming from the Node Properties Panel

The transform handles are a convenient way to move objects around in the 3D workspace, but when you want more precision, you should enter values directly into the object's node panel. The panel also includes transform and rotation order options, which are not available within the 3D Viewer.

The following assumes you've already loaded the object's parameters into the Properties Bin.

To set transformation options:

- From the **transform order** list, select an option to define the order by which transformations are executed (**s** signifies scale, **r**, rotation; and **t**, translation).
- From the **rotation order** list, select an option to define the axial order by which rotations are executed.

To transform an object from its panel:

- To move the object along one or more axes, increment or decrement the **translate x**, **y**, and **z** fields.
- To rotate the object, increment or decrement the **rotate x**, **y**, and **z** fields.
- To scale the object on all axes simultaneously, increment or decrement the **uniform scale** field.
- To scale the object asymmetrically (on x, y, or z), increment or decrement the **scale x**, **y**, and **z** fields.
- To skew the object (warp it by rotating its local axes), increment or decrement the **skew x**, **y**, and **z** fields to rotate the corresponding axis (and associated object vertices).

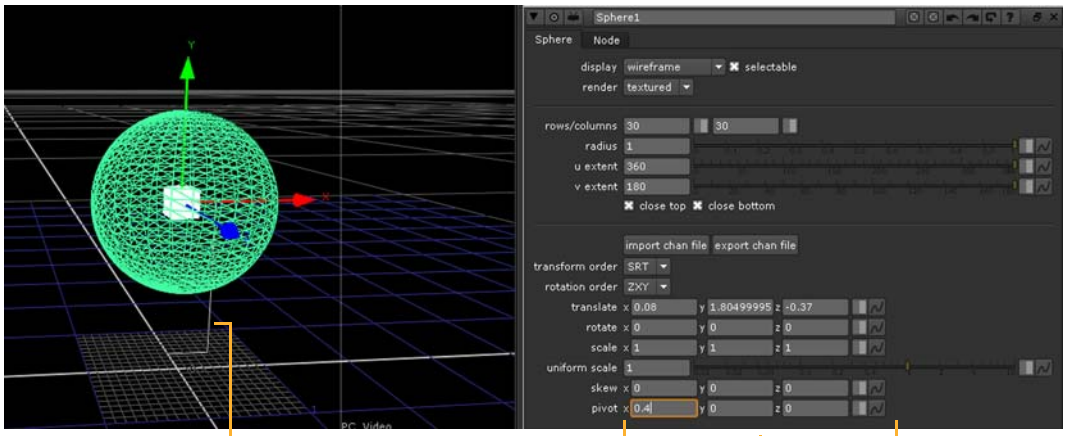
Transformations and the Pivot Point

When you make changes to an object's position, scaling and rotation, these occur from the location of the object's origin point or *pivot*. By default, the pivot point is located at the intersection of the object's local axes.

You can offset the pivot point and move it anywhere you like—you can even move it outside of the object. Subsequent local transformations will then occur relative to the new pivot point location.

To move the pivot point:

1. Double-click on the object node to display its parameters.
2. Change the values of the **pivot x**, **y**, and **z** fields to move the local axis in any direction.

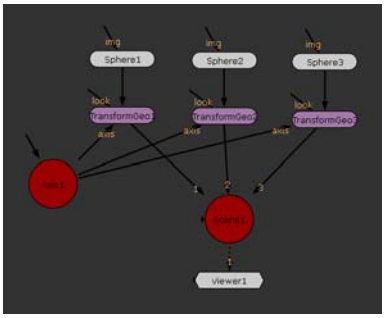


The object's graphical overlay points to the location of the pivot point with a line. All subsequent local transformations occur relative to this pivot point.

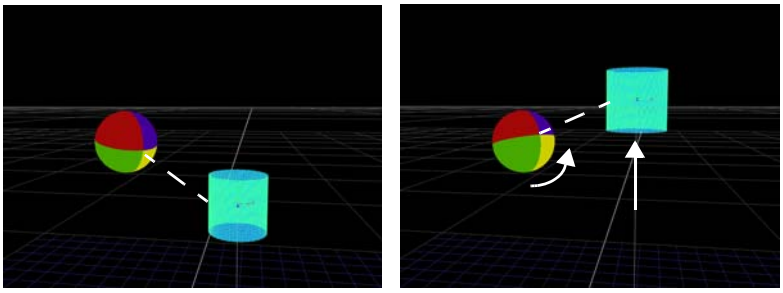
Once you've defined the location of an object's pivot point, you can use the object's transform parameters to translate, rotate, scale, and skew the object relative to the pivot point.

Using the TransformGeo Node

The TransformGeo node allows you to move, rotate, scale, and perform other transformations on several objects merged together with a MergeGeo node. It also lets you connect geometry objects to an Axis node. By doing so, you can move all the connected objects together by using the Axis transformation controls. All you need to do is insert a TransformGeo after each geometry object, connect the Axis node to the TransformGeo nodes' **axis** input, and adjust the transform controls of the Axis node. For more information, see *Parenting to Axis Objects* on page 288.

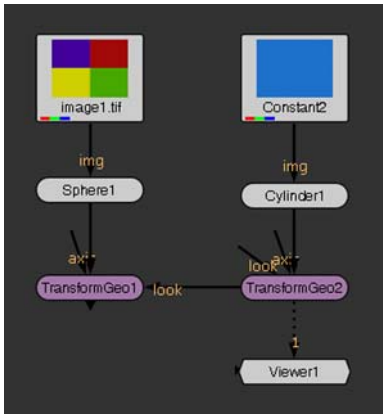


Another use of the TransformGeo node is to have the rotation of one object depend on the position of another so that the first object is always rotated to face or "look at" the second one. For example, you can have a sphere object always facing a cylinder object, regardless of the cylinder's position. When the cylinder is moved to a new position, the sphere is automatically rotated to face it.

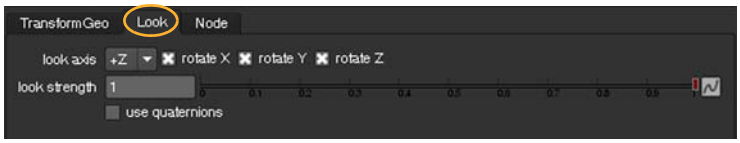


To have one 3D object always face another:

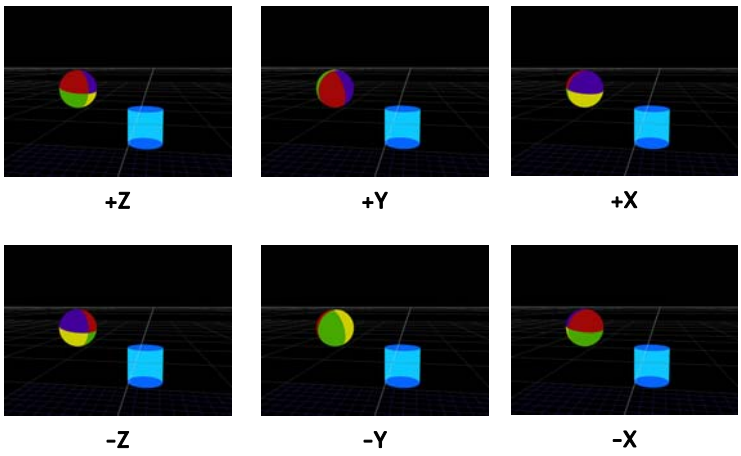
1. Select the 3D object node (for example, a sphere) that you want to face another object.
2. Choose **3D** > **Modify** > **TransformGeo** to insert a TransformGeo node.
3. Select the object you want the first object to face (for example, a cylinder), and insert a TransformGeo after this node, too.
4. Connect the first TransformGeo node into the **look** input of the second TransformGeo node.



5. Open the controls of the first TransformGeo node and go to the **Look** tab.



6. From the **look axis** pulldown menu, select the axis around which the object will be rotated to face the other object:



7. Use the **rotate X**, **rotate Y**, and **rotate Z** check boxes to select the axes the object will be rotated around. For the first object to truly face the second, you need to check all three check boxes.
8. Adjust the **look strength** slider to define the extend of the rotation. The smaller the value, the less the object is rotated. Setting the value to 0 produces no rotation.
9. If you want to use an alternate scheme to calculate the rotation, check **use quaternions**. This may be useful for smoothing out erratic rotations along the selected **look axis**.

If you now adjust the second TransformGeo node's transform controls, you'll notice that the first object automatically rotates to face the second object. For more information on how to adjust the transform controls, see *Using the Transform Handles* on page 298 and *Transforming from the Node Properties Panel* on page 299.

Modifying Object Shapes

Many nodes under the Modify menu let you modify the shape of an object as a whole. Modifying only selected portions of an object is currently not supported.

You can modify 3D objects using lookup curves, power functions, images, a Perlin noise function, a distortion function, and a trilinear interpolation.

Modifying Objects Using Lookup Curves

The CrosstalkGeo and LookupGeo nodes offer you direct global control over each of the vertex x, y, and z values respectively. You can, for example, only modify all the y values without touching the x and z values.

You change the different vertex values (x, y, or z) by modifying their associated 2D curves in lookup tables (LUTs). The x axis in the LUT represents the current vertex value, and the y axis the new vertex value.

By default, the curve is a diagonal line where all the points in the curve have the same value on the y axis (the new value) as they do on the x axis (the current value). Because both x and y values are the same, there is no change in the object's shape.

By modifying, for example, the CrosstalkGeo node's **y** LUT the following way, you can set some of the vertex y values of a sphere to 0 to squash its bottom half:

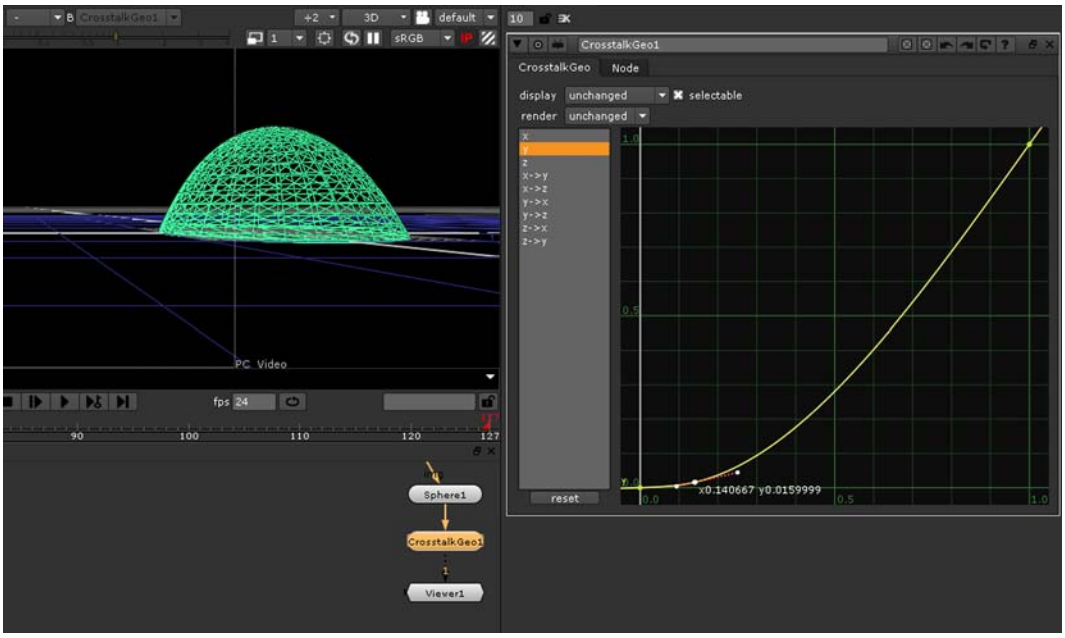


Figure 13-18: Modifying the CrosstalkGeo node's LUT.

With the CrosstalkGeo node, you can also use one of the vertex x, y, and z values to evaluate the lookup curve and then add the result to another vertex value. For example, you could modify the x->y curve, using the vertex x value to find the new value on the curve, and then add that to the vertex y value. This way, you can modulate the y values by another channel.
By default, these curves are horizontal lines at y=0. They produce no change, because the value added to the vertex (the new value on the y axis) is 0.

To modify objects using lookup curves:

1. Select **3D** > **Modify** > **CrosstalkGeo** or **LookupGeo** to insert a CrosstalkGeo or LookupGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it.
4. From the list on the left, select the curve you want to modify. For example, you'd select **z** to only modify the vertex z values.

In the case of the CrosstalkGeo node, you can also select **y->x**, for example, to use the vertex y value to evaluate the curve and add the result to the vertex x value.

5. Adjust the curve as necessary. To insert points on the curve, **Ctrl/Cmd**+**Alt**+click on the curve.

Modifying Objects Using a Power Function

The LogGeo node lets you modify the shape of your 3D objects using a power function. Using this node, you can raise each of the vertex x, y, and z values to a power (X^x, Y^y, Z^z). This can have a different effect depending on whether you are dealing with negative or positive values.

To modify objects using a power function:

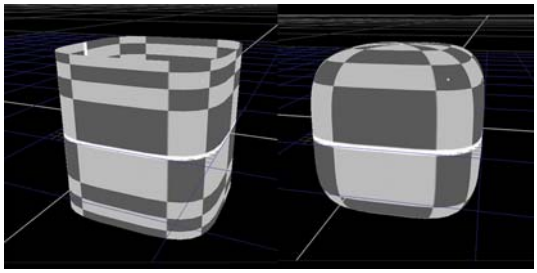
1. Select **3D** > **Modify** > **LogGeo** to insert a LogGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it. See *Object Display Properties* on page 296.
4. Check **swap**. This swaps the values and the powers they are raised to around (for example, changes 5^7 into 7^5).
5. In the **log x**, **y**, and **z** fields, enter the power you want to raise the respective vertex values to. For example, if you want to raise the vertex z values to the power of 20, enter 20 in the **z** field.

   Alternatively, you can adjust your 3D object in the Viewer by dragging the white control point to a new location. You can find the control point just outside the object.
6. To clamp the negative x, y, and z values to 0.0, check **clamp black**. This option is only valid if you have checked **swap**.

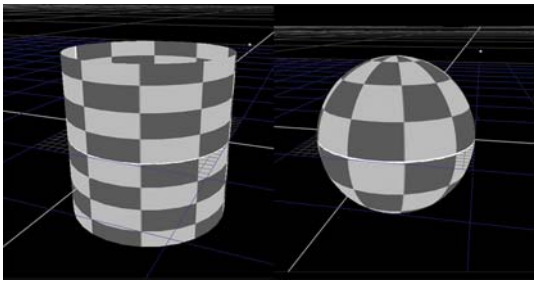
Tip

If you set the **log x**, **y**, and **z** values to 1 and check **swap**, the LogGeo node produces no change in the incoming geometry. If you want to try out how the node works, this is a good place to start as you can then gradually adjust the values from there.

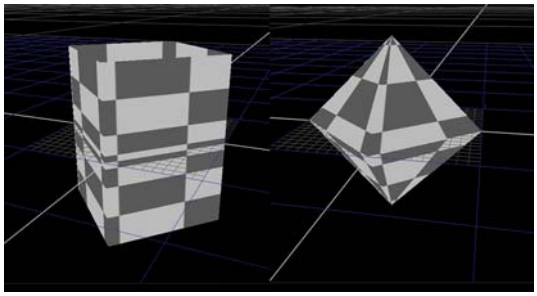
The following images illustrate the effect of the LogGeo node on the default Nuke cylinder and sphere when **swap** is checked in the LogGeo controls. Notice that if these objects were not positioned in the default location (centered around 0,0,0), the results would be different.



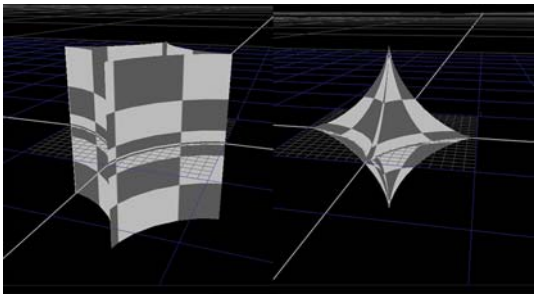
Log x, y, and z set to 0.5.



Log x, y, and z set to 1 (no change).



Log x, y, and z set to 2.



Log x, y, and z set to 3.

Figure 13-19: The LogGeo node applied to the default cylinder and sphere.

Modifying Objects Using an Image

With the DisplaceGeo node, you can modify geometry based on an image. When using the node, each vertex is displaced along its normal with a value corresponding to the image pixel the vertex's uv attribute points to. The higher the pixel value, the greater the displacement. The following image illustrates the principle behind the DisplaceGeo node. A Card node is modified to resemble the pattern of a Checkerboard image.

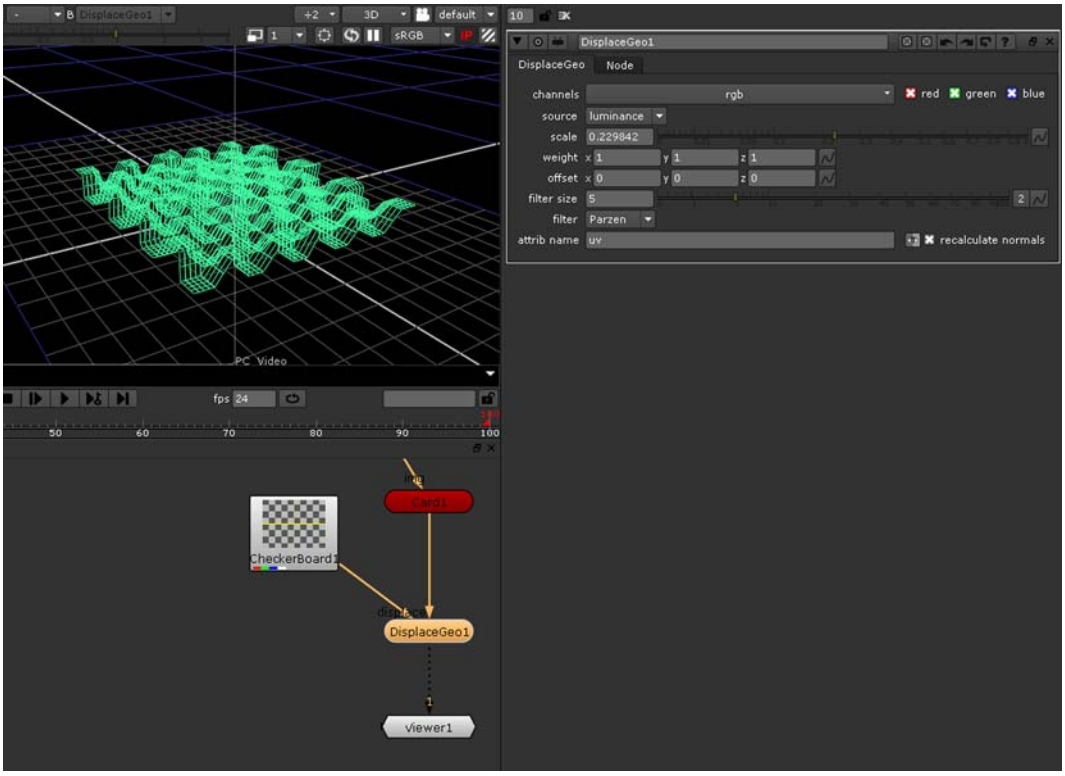


Figure 13-20: Using the DisplaceGeo node to modify geometry.

To modify objects using an image:

1. Select **3D** > **Modify** > **DisplaceGeo** to insert a DisplaceGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it.
4. Read in your image map and connect it to the DisplaceGeo node's **displace** input.
5. Adjust the following controls:
   - From the **channels** pulldown menu and check boxes, select the channels to use for the displacement value.
   - From the **source** pulldown menu, select the source for the displace value. For example, if you selected **rgb** or **rgba** from the **channels** pulldown menu, you can use the red, green, or blue channel or the pixel luminance as the source.
   - To define the scale of the displacement, adjust the **scale** slider. The higher the value, the bigger the displacement.
   - To give x, y, and z different weightings, enter new weights the **weight** fields. By default, each weighting is set to 1. If you don't want to make changes to a value, set its weight to 0.

- To offset x, y, and z values, enter the value by which you want to offset them in the **offset** fields. For example, if you enter 0.5 in the y offset field, 0.5 is added to the y value.
- To change the size of the filtering applied to the image before the displacement, adjust the **filter size** slider.
- To select the filtering algorithm applied to the image before the displacement, select an algorithm from the **filter** pulldown menu. For more information, see *Choosing a Filtering Algorithm* on page 160.
- To change the name of the attribute that's used as the vertex's UV coordinates to find the image pixel, enter a name in the **attrib name** field.
- Usually, the normals aren't correct after the vertices have been moved. To recalculate them after the displacement, check **recalculate normals**.

Modifying Objects Using a Perlin Noise Function

The ProcGeo node lets you modify your 3D objects using a Perlin noise function that creates *seemingly* random noise. For example, you could use the ProcGeo node to generate animated noise for rippling waves or clouds, or to create a terrain from a flat card, like in the following image:

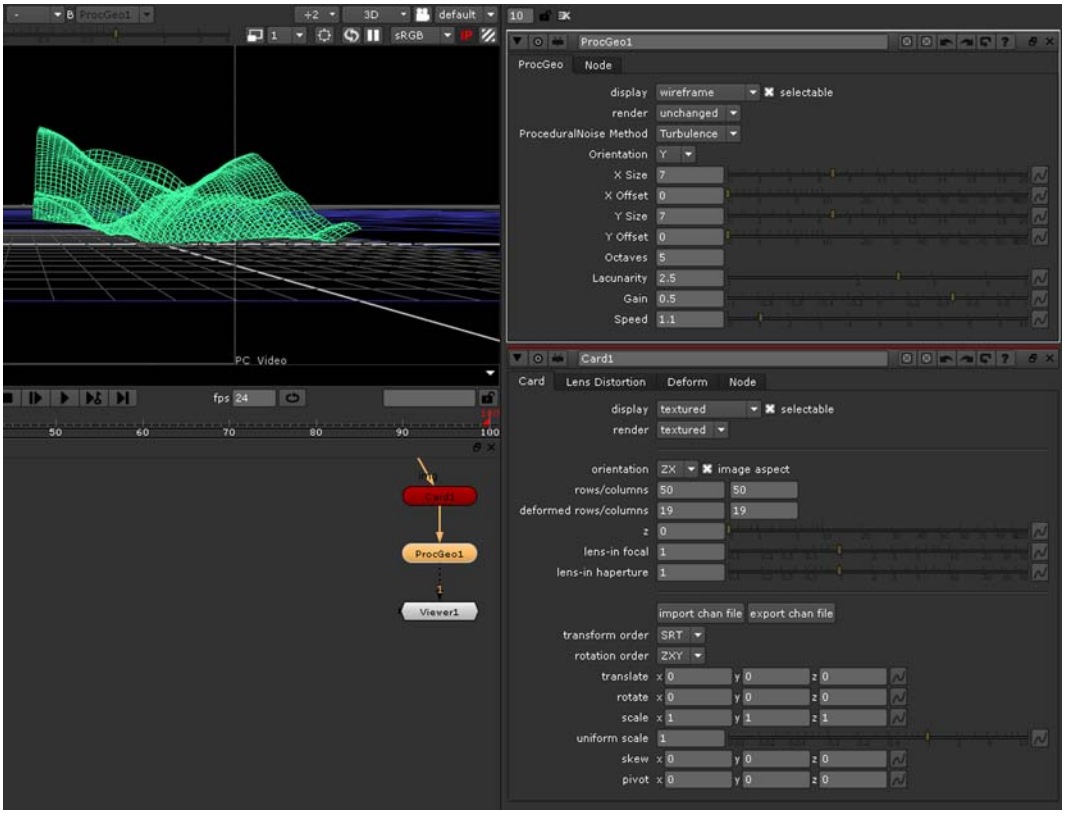


Figure 13-21: Using the ProcGeo node to create a terrain from a card object.

You can select the type of noise and control its look in the ProcGeo node's parameters.

To modify objects using a Perlin noise function:

1. Select **3D** > **Modify** > **ProceduralNoise** to insert a ProcGeo node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it.
4. From the **ProceduralNoise Method** pulldown menu, select the type of noise you want to use: **Turbulence** or **fBm** (Fractal Brownian Motion).
5. To select whether to modify the x, y, or z values or all of them, use the **Orientation** pulldown menu.
6. To change the look of the noise, adjust the rest of the parameters. For example, to control the amount of detail of the noise, adjust **Octaves**.

Modifying Objects Using a Distortion Function

The RadialDistort node is a non-linear transformation of the vertices along directions from the object centre, giving either a barrel or pin-cushion distortion. In the following image, two cylinders have been distorted using the RadialDistort node.

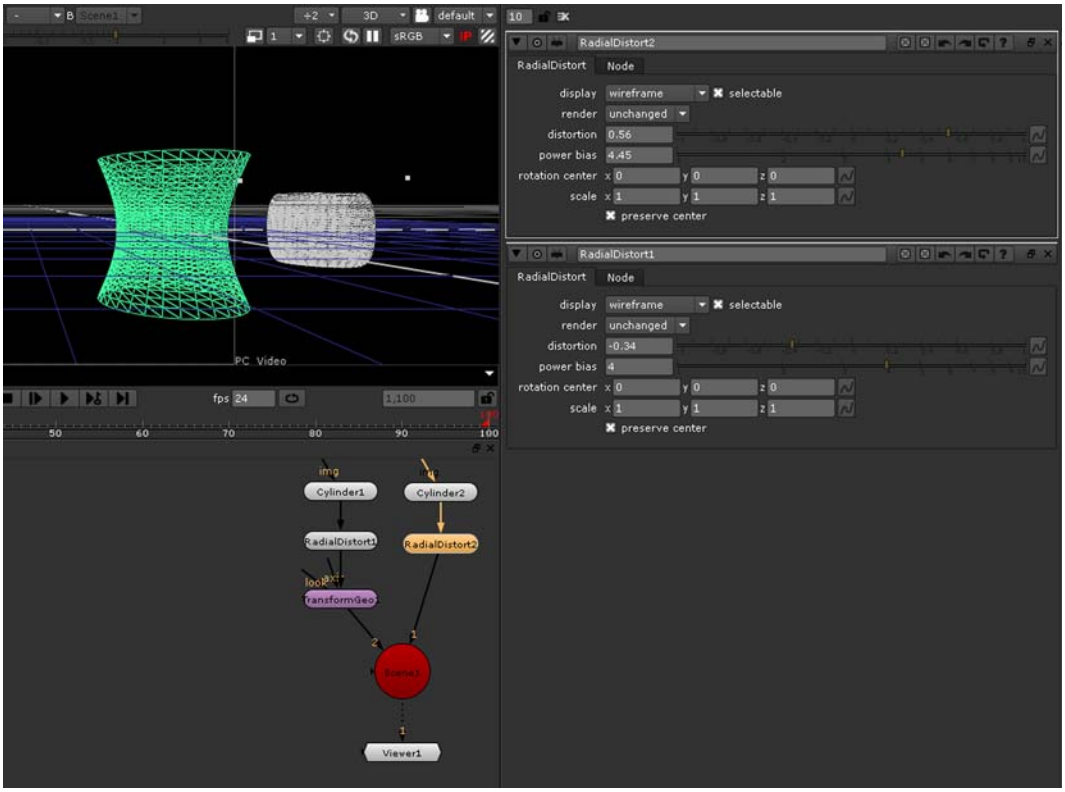


Figure 13-22: Barrel and pin-cushion distortions.

To modify objects using a distortion function:

1. Select **3D** > **Modify** > **Radial Distort** to insert a RadialDistort node anywhere after the 3D object you want to modify.

2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it.
4. To select whether the distortion is a barrel or pin-cushion, adjust the **distortion** slider. Values below 0 produce a barrel distortion, whereas values above 0 produce a pin-cushion distortion. If you set the value to 0, the 3D object is not distorted.
5. To control the magnitude of the distortion, adjust the **power bias** slider. The higher the value, the more distorted the object becomes.
6. To move the centre point of the distortion, enter new coordinates in the **rotation center** fields.
7. To control the amount of distortion in each of the x, y, or z directions, adjust the values in the **scale** fields.
8. To keep the object's centre in its original place in the 3D space, check **preserve center**.

Modifying Objects Using a Trilinear Interpolation

With the Trilinear node, you can warp the object as a whole by using a trilinear interpolation to warp the object's bounding box. For example, you can use this node to create animated object deformations, such as the squish/squash of a bouncing ball.

To modify objects using a trilinear interpolation:

1. Select **3D** > **Modify** > **Trilinear** to insert a Trilinear node anywhere after the 3D object you want to modify.
2. Attach a Viewer to the node to see your changes.
3. In the node's controls, use the **display** pulldown menu to select how you want to view your object in the Viewer while making changes to it.
4. To move each corner of the bounding box, enter new coordinates in the **p0**, **p1**, **p2**...**p7** fields. To cancel your changes and reset the box, select **reset shape to input**.
5. To not use the object's bounding box but define a box yourself, go to the **Source box** tab and uncheck **use incoming bounding box**. Adjust the **src0** and **scr1** coordinates define the box. To change the colour of the box, click the **box** button.

Lighting

The nodes under the Lights menu let you control the lighting in your scene. Using these nodes, you can bring objects out or push them back, create an illusion of depth, simulate the conditions in the real world, or simply alter the feeling of the scene.

Nuke features four types of light you can use in your 3D scenes: direct light, point light, spot light, and environment light. You can add these using the DirectLight, Point, Spotlight, and Environment nodes.

In addition to the nodes mentioned above, there is a Light node, which lets you read in lights from .fbx files (for more information, see *Importing Channel Files, Cameras, Lights, Transforms, and Meshes from Other Applications* on page 321).

The Light node also includes the DirectLight, Point, and Spotlight nodes, so you can set it to act as any of these three nodes. Simply insert a Light node (select **3D** > **Lights** > **Light**) and

choose the light type you want to use. The advantage of using a Light node in this way is that if you want to change the light type later, you can do so without setting up a new node. For example, you might insert a direct light, but then realise that what you actually need is a spot light. If you inserted the direct light using a DirectLight node, you need to delete this node and insert a SpotLight node instead. However, if you inserted the direct light using a Light node, you can simply change the light **type** from **directional** to **spot** in the Light controls.

Direct Light

A direct light is a light that emits parallel light in one direction. It appears to illuminate all objects with equal intensity, as if it was coming from a far away source. Being at an infinite distance from the objects, direct light has orientation, but no position. A real world example of a direct light is the sun. You can use direct light to simulate sunlight and moonlight, for example.

To add a direct light:

1. Select **3D** > **Lights** > **Direct** to insert a DirectLight node in your script.
2. Connect the DirectLight node to the Scene node.
3. In the DirectLight node's controls, adjust the following:
   - Drag the **color** slider to change the light color.
   - Drag the **intensity** slider to change the brightness of the light.
   - To control the direction of the light, enter values in the **rotate** fields.

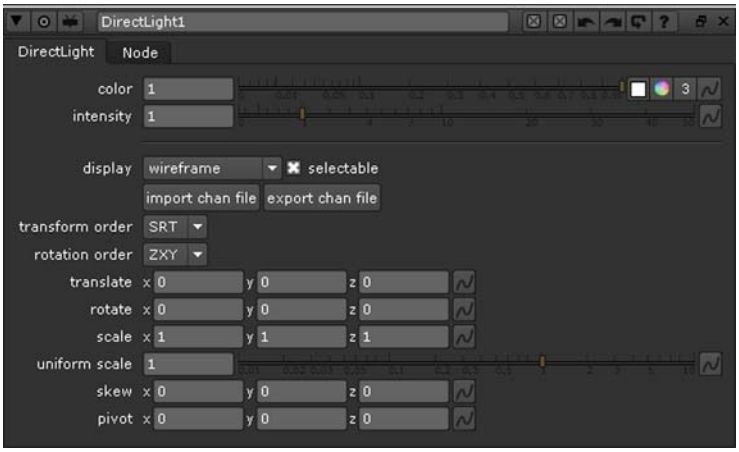


Figure 13-23: Direct light controls.

Point Light

A point light is a point in 3D space that emits light in every direction. A real world example of a point light is a light bulb. You can use point light to simulate light bulbs, lamps, and candles, for example.

To add a point light:

1. Select **3D** > **Lights** > **Point** to insert a Point node in your script.
2. Connect the Point node to the Scene node.

3. In the Point node's controls, adjust the following:
    - Drag the **color** slider to change the light colour.
    - Drag the **intensity** slider to change the brightness of the light.
    - To control how much light the object gets from the light source (based on the distance between the object and the light source), use the **falloff type** menu. A **Linear** type diminishes the light at a fixed rate as it travels from the object, whereas **Quadratic** and **Cubic** types diminish the light at an exponential rate. If you select **No Falloff**, the distance between the light source and the object does not affect the lighting.
    - To control the position of the light in the 3D space, enter values in the **translate** fields.

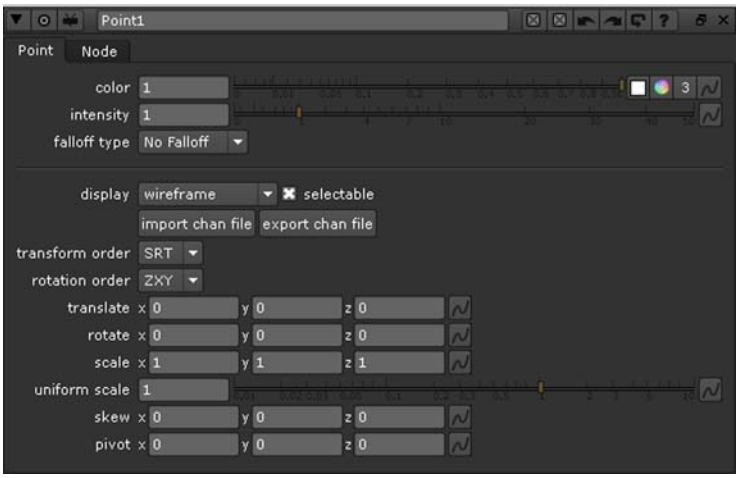


Figure 13-24: Point light controls.

Spot Light

A spot light is a point in 3D space that emits a cone-shaped light in a given direction. A real world example of a spot light is a desk lamp.

To add a spot light:

1. Select **3D** > **Lights** > **Spot** to insert a Spotlight node in your script.
2. In the node's controls, adjust the following:
    - Drag the **color** slider to change the light colour.
    - Drag the **intensity** slider to change the brightness of the light.
    - Drag the **cone angle** slider to control the spread of the light (how wide or narrow the beam is) in degrees from 0 to 180.
    - Drag the **cone penumbra angle** slider to control the softness along the edge of the area of illumination. A negative value fades inward from the circle's edge. A positive value fades outward from the circle's edge. The **cone falloff** should be set to zero or a low value in order to see the softness. This feature is only visible in the rendered objects and not in the 3D OpenGL Viewer.

- Drag the **cone falloff** slider to control how concentrated the light is (that is, how much the light diminishes from the centre of the circular region out to the edge). The higher the value, the more focused the light becomes. The falloff is independent of the **falloff type**.
- To control how much light the object gets from the light source (based on the distance between the object and the light source), use the **falloff type** menu. A **Linear** type diminishes the light at a fixed rate as it travels from the object, whereas **Quadratic** and **Cubic** types diminish the light at an exponential rate. If you select **No Falloff**, the distance between the light source and the object does not affect the lighting.
- To control the direction of the light, enter values in the **rotate** fields.
- To control the position of the light in the 3D space, enter values in the **translate** fields.

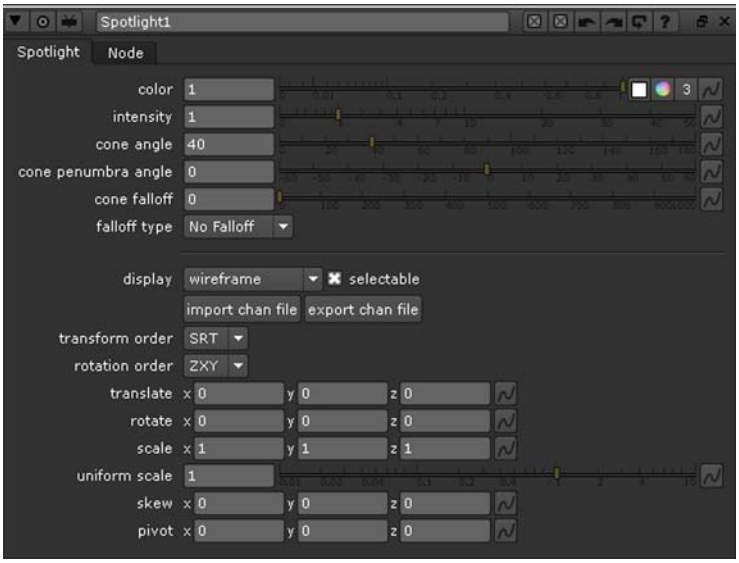


Figure 13-25: Spot light controls.

Environment Light

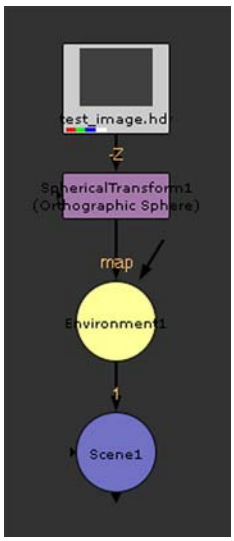
An environment light is a light that illuminates the objects using an image of light from a real-world environment. This image-based lighting is generated using High Dynamic Range Images (HDRI). When HDR images are created, several differently exposed images are combined to produce a single image of the surrounding environment. As a result, HDR images have a wide range of values between light and dark areas, and represent the lighting conditions of the real world more accurately.

To use environment light, you first need to shoot a real life environment as an HDR image. Using the Environment Maps node, you then convert this image into a spherical mapped image. The sphere is used to surround the 3D objects, so that the mapped image colour illuminates them.

Environment light only works with shiny object materials that can reflect the mapped image. It results in a very realistic lighting that makes it easier to integrate the objects into the environment.

To add an environment light:

1. Read an HDR image of the environment into your script.
2. Select **Transform** > **Environment Maps** to insert a SphericalTransform node after the HDR image. You use this node to convert the HDR image into a spherical mapped image. In the nodes controls, select the **Input Type** and the **Output Type** (in this case, **Sphere**).
3. Select **3D** > **Lights** > **Environment** to insert an Environment node in your script. Connect the SphericalTransform node to the Environment node's **map** input, and the Environment node to the Scene node.



4. In the Environment node's controls, adjust the following:
   - Drag the **color** slider to change the light colour.
   - Drag the **intensity** slider to change the brightness of the light.
   - From the **filter** pulldown menu, select a filtering algorithm for the map image. For more information, see *Choosing a Filtering Algorithm* on page 160.
   - To change the blur size of the map image, adjust the **blur size** slider.

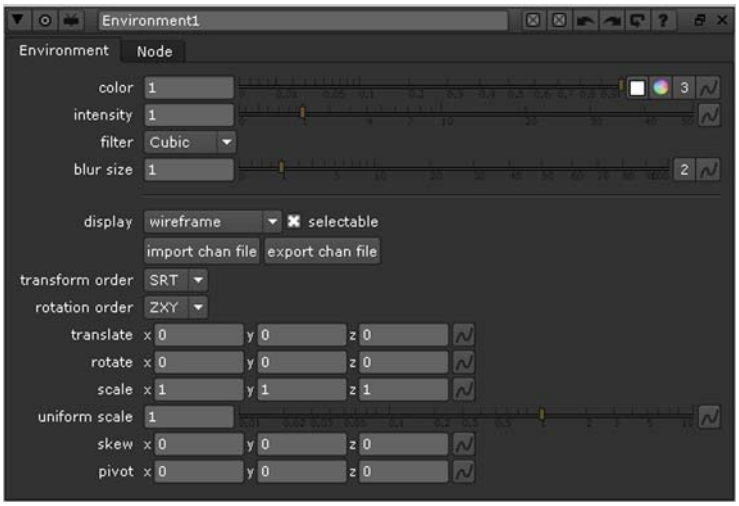


Figure 13-26: Environment light controls.

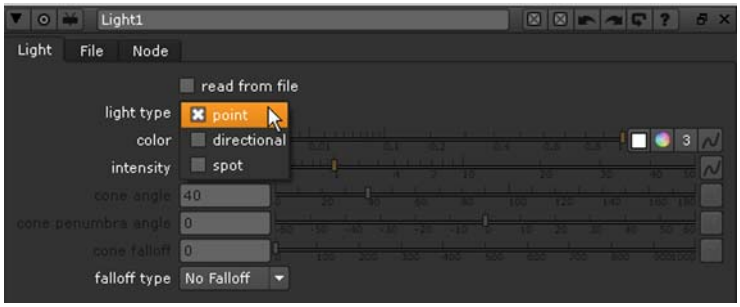
The Light Node

You can use the Light node to add a direct light, point light, or spot light into your script.

The node can also be used to import lights from .fbx files. This is described later, under *Importing Channel Files, Cameras, Lights, Transforms, and Meshes from Other Applications* on page 321.

To add a direct, point, or spot light:

1. Select **3D** > **Lights** > **Light** to insert a Light node into your script.
2. In the Light controls, select the **light type** you want to use: **point**, **directional**, or **spot**. The controls are enabled and disabled according to the light type you choose. For example, if you chose directional light, you get the same controls that appear on the DirectLight node.



3. Adjust the controls as necessary. For information on the functions of the controls, refer to the following:
   - If you selected point as the light type, see *Point Light* on page 310.
   - If you selected directional as the light type, see *Direct Light* on page 310.
   - If you selected spot as the light type, see *Spot Light* on page 311.

Manipulating Object Normals

Object normals are vectors that are perpendicular to the surface. They are used in lighting calculations to determine how the light should bounce off a surface at any particular point. By manipulating them, you can control the diffuse and specular light contributions.

To manipulate object normals:

1. Select **3D** > **Modify** > **Normals** to insert a Normals node anywhere after the 3D object whose lighting you want to adjust.
2. Connect a Camera, Axis, or light node to the Normals node's **lookat** input.
3. In the Normals controls, open the **action** pulldown menu and select:
   - **unchanged** to make no changes.
   - **set** to assign the normals value to the normal x, y, and z fields.
   - **build** to rebuild each normal based on the surrounding vertices. Adjust the **threshold angle** slider to determine the break angle where two faces no longer constitute a smooth surface. An angle of 0 means all faces are flat, whereas 180 means all faces are smooth. A good average setting is 60.
   - **lookat** to point all normals towards the Normals node's **lookat** input.
   - **delete** to remove the named attribute from the object. For example, if you remove the N attribute, the object has no normals.

Working with Cameras

Nuke lets you add multiple cameras to a scene, with each providing a unique perspective. You can also setup cameras that project 2D images onto geometry in your scene.

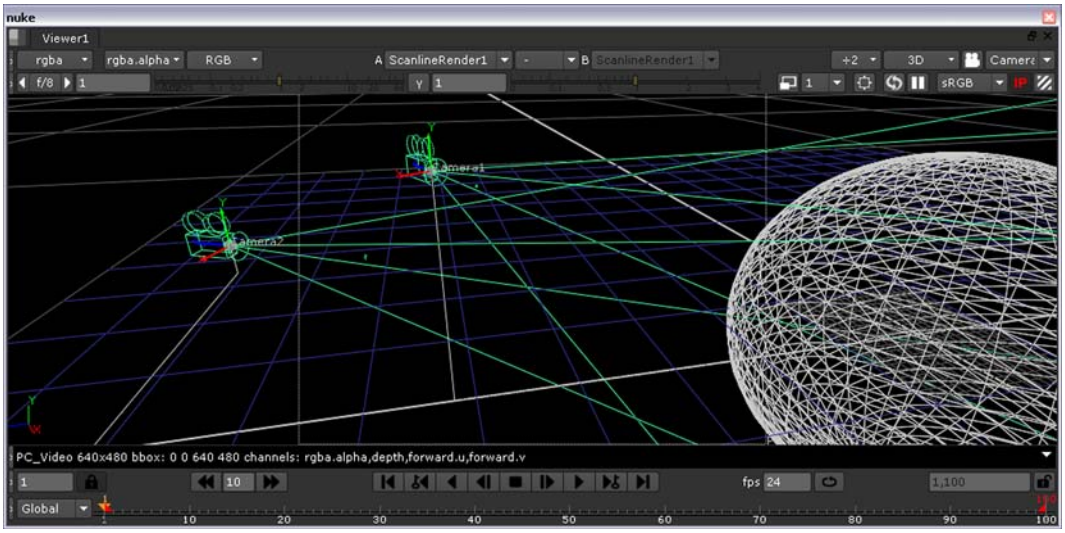


Figure 13-27: Cameras in the scene.

To add a camera:

1. Click **3D > Camera** to insert a Camera node.

2. To setup the rendering camera, drag a connector from the new Camera node to the ScanlineRender node.

   or

   To setup an additional scene camera for viewing, drag a connector from the new Camera node to the Scene node.

To "see" the scene through a particular camera, you need to select the camera and press **H**. Make sure you're in the 3D perspective view (press **V** before **H**) before trying this; it doesn't work when you're looking at your scene through an orthographic view.

To edit a camera's lens characteristics:

1. If necessary, double-click on the Camera node to display its parameters.
2. Click the **Projection** tab.
3. Drag the **focal length** slider to adjust the camera's level of magnification.
4. Drag the **near** slide to edit the position of the camera's forward clipping plane. Objects in front of this plane will not be rendered or displayed.
5. Drag the **far** slider to edit the position of the camera's rearward clipping plane. Objects in behind this plane will not be rendered or displayed.
6. Increment the **window translate u** (horizontal axis) and **v** (vertical axis) sliders to translate the camera's output along either axis.
7. Increment the **window scale u** (horizontal axis) and **v** (vertical axis) sliders to scale the camera's output on either axis.
8. Drag the **window roll** slider to rotate the camera's output on z.

Projection Cameras

In addition to viewing and rendering a 3D scene, cameras can also project a 2D still image or image sequence onto geometry in the scene. This is similar to the front-projection systems used in practical photography, where a background image or other element is projected onto the stage and photographed with other elements.

In Nuke, a projection camera can receive camera data tracked from the original shot—or another shot—to setup a projection that is match-moved to another source.

This setup requires these nodes: a projection camera, a Scene node, a Project3D node, a geometry object node (what you'll be projecting onto), and a 2D node with the image that you want to project.

First a Little Math...

When you create a projection camera, you need to gather some information and do a few small calculations to make sure the projection works. Here are the bits of information you need:

- Focal length of the lens that photographed the projection image.
- Resolution of scanned image.
- Scanner pitch of the film scanning device.

After you have this information, you need to do these calculations to get the horizontal and vertical aperture settings for the projection setup:

```
horiz. res. / scanner pitch = horizontal aperture
vertical res. / scanner pitch = vertical aperture
```

So, for example, if your image resolution is 720 x 486 and the scanner pitch is 20, then these are the results:

```
720 / 20 = horizontal aperture = 36
486 / 20 = vertical aperture = 24.3
```

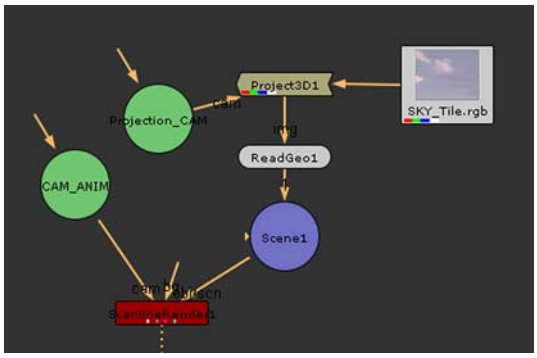
Generally, for most professional projects, you can get the lens focal length from the camera report for the shot(s). If that is not available, you may be able to extrapolate lens information by running the shot through a 3D tracking application, such as Boujou, Syntheyes, or RealViz.

Setting Up the Projection Camera Script

Once you have the horizontal and vertical aperture and the lens focal length for the image you want to project, you can complete the projection camera setup.

To add a projection camera:

1. Choose **3D > Camera** to add a new camera to your script and rename the node to identify it as a projection camera.
2. Choose **3D > Shader > Project3D** to add a Project3D node to the script.
3. Connect the 2D image (i.e., Read node) to the Project3D node.
4. Connect the projection camera to the Project3D node.
5. Connect the Project3D node to the geometry node that should receive the 3D projection.
6. Double-click the projection camera node to load its parameters.



7. Click the Projection tab in the camera's panel and then enter the information you gathered for **focal length**, **horiz aperture**, and **vert aperture**.

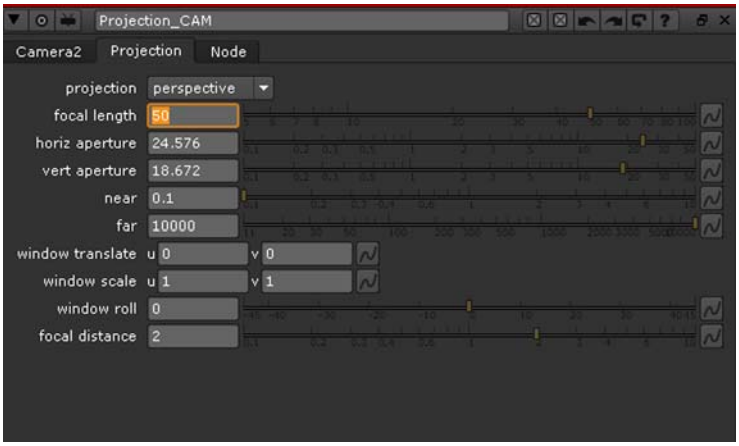


Figure 13-28: Entering projection camera settings.

When you are finished, view the 3D scene to check the placement of the projection. The next section explains how to preview the 2D and 3D elements together, to check the results of the composite.

To view a 3D scene over a 2D background image:

1. Select the Scene node and press **1** to display its output to the Viewer.
2. If necessary, press **Tab** to toggle the Viewer to 3D mode.
3. Select the rendering camera object or node and press **H** to look through it.
4. Select the node with the 2D image you want to see in the Viewer, and then press **Shift+2**.

   The Shift+2 keystroke connects the image to the Viewer (assigning the next available connection, number 2), and also sets up the compare wipe.
5. Choose the desired option from the Viewer composite list (i.e., **–** (none), **over**, **under**, **minus, wipe**).

This last step superimposes the two elements in the Viewer. The crosshair (shown below in Figure 13-46) is the control that lets you adjust the location and angle of the wipe for the comparison.

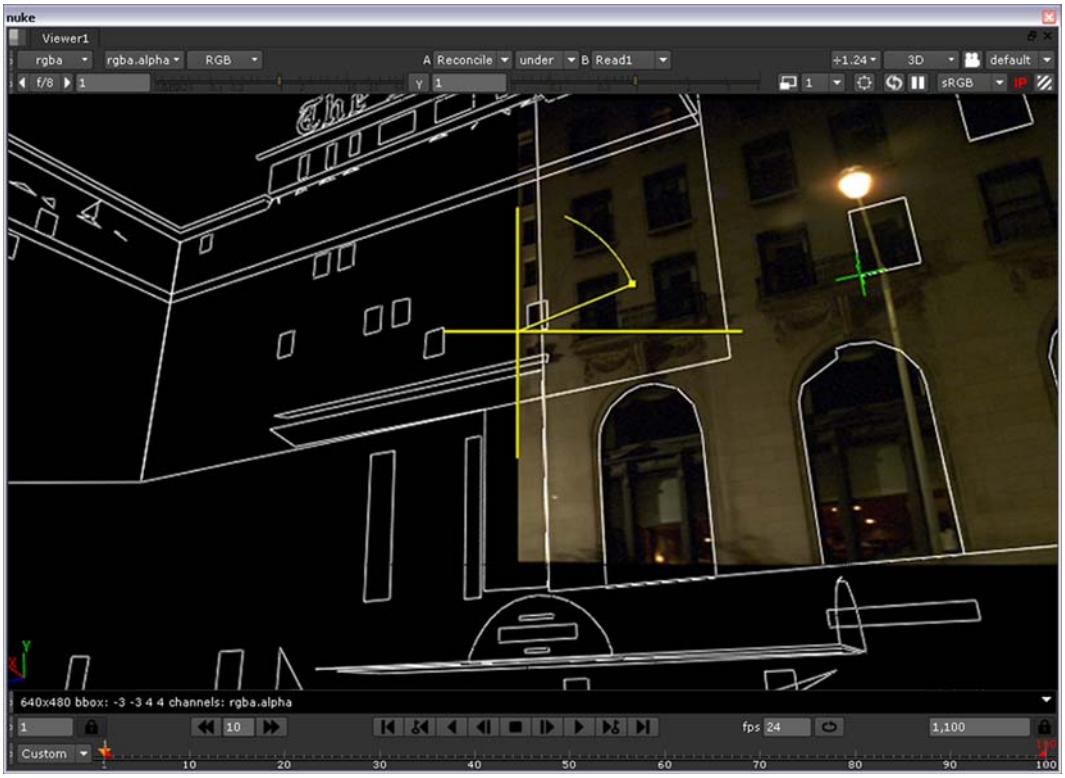


Figure 13-29: Comparing a 3D scene over a 2D image.

Adding Motion Blur to the 3D Scene

To create more realism for a 3D scene, you'll want to add motion blur to it based on the movement of your 3D camera. This can be done in two ways:

1. If you have moving objects in your 3D scene or the camera movement over the shutter time is non-linear, adjust the **sample** value in the ScanlineRender nodes parameters. The image is sampled multiple times over the shutter period. This way is the most accurate, but also the slowest, because the full rendering calculation is done multiple times for each pixel.
2. If your 3D scene is static or nearly so and the camera movement over the shutter time is nearly linear, add the MotionBlur3D and VectorBlur nodes after the ScanlineRender node in your script. This way is faster to render.

To add motion blur for a scene with moving objects or non-linear camera movement during the shutter period:

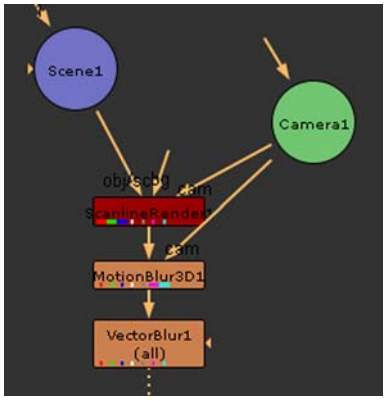
1. In the ScanlineRender node's controls, go to the **MultiSample** tab.
2. Increase the **sample** value to sample the image multiple times over the shutter period.

Tip

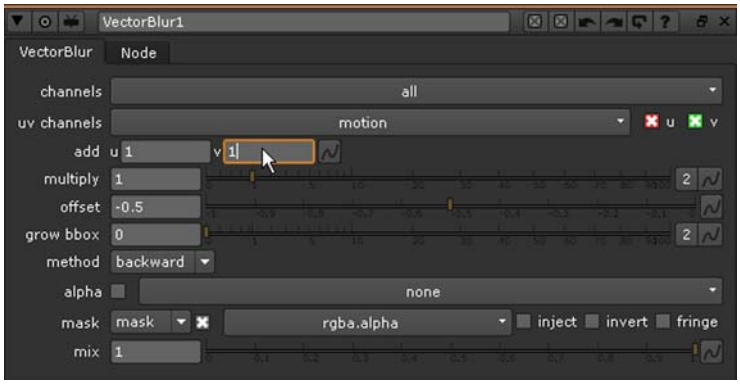
By increasing both the **sample** and **focus diameter** values in the ScanlineRender node's controls, you can also use the multisampling to simulate depth of field.

To add motion blur for a static scene with approximately linear camera movement during the shutter period:

1. Select the **ScanlineRender** node.
2. Choose **Filter** > **Motion Blur 3D** to insert this node and connect it to the scanline renderer.
3. Connect the rendering camera to the MotionBlur3D node.
4. Choose **Filter** > **Vector Blur** to insert and connect this node to the MotionBlur3D node.



5. In the VectorBlur settings, select the **motion** layer from the **uv channels** list.

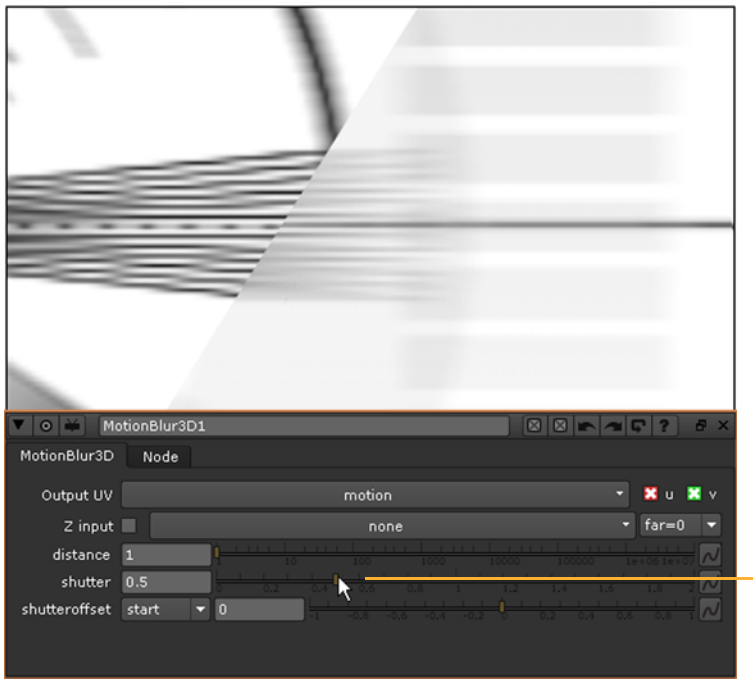


6. For the **add** settings, enter 1 for both **u** and **v**.

Note

For anamorphic footage, the add u value should be 2 and the add v value should be 1.

7. To adjust the length of the blur, adjust the **Shutter** setting in the MotionBlur3D properties panel.



Importing Channel Files, Cameras, Lights, Transforms, and Meshes from Other Applications

Sometimes, you may need to import files or objects created in 3D applications, such as Maya or Boujou. Depending on what you want to import and from where, there are different ways of doing the import:

- To apply motion data calculated with 3D tracking software to cameras or objects, you need to import channel files. For more information, see *Applying Tracks to an Object* below. You can also use channel files to import cameras created in other applications into Nuke. However, as the chan file format is not a standard file format, you may need a file format converter to export chan files from other applications.
- To import cameras, lights, transforms, or meshes from other applications, you can use .fbx files. Fbx is a standard file format many applications can export to. Fbx files contain 3D scenes from which you can import cameras, lights, transforms, and meshes into Nuke. For more information, see *Working with .fbx Files* on page 322.
- To import cameras from Boujou, you can use the import_boujou.tcl script shipped with Nuke. For more information, see *Importing Cameras from Boujou* on page 326.

Applying Tracks to an Object

Nuke can import channel files and apply the motion data to the transformation parameters of any camera or object. The most common purpose for this is to simulate a practical camera move or move objects along a defined path.

Channel files contain a set of cartesian coordinates for every frame of animation in a given shot. This information is calculated by 3D tracking software, such as 3D-Equalizer, Maya, or Boujou, and then exported as channel files.

To apply a channel file to an object:

1. Double-click on an object or camera node to display its parameters.
2. Click **import chan file**. The file navigation dialog appears.
3. Navigate to the channel file, then click **OK**.
4. Nuke reads in the channel data and displays a status message about the number of data frames imported. You'll also notice the object's translation parameters turn green to indicate these parameters are now controlled by animation data. Scrub the Viewer and you'll notice the object or camera now moves according to the transformation data imported from the channel file.

Note

You can use the **export chan file** button to export as a chan file any animated translation parameters which you've applied to given object. This is a useful method of sharing setups between artists.

Working with .fbx Files

Fbx is a standard 3D file format that gives you access to 3D scenes created in other applications supporting the same format. What you generally have in an .fbx file is an entire 3D scene containing cameras, lights, meshes, non-uniform rational B-spline (NURBS) curves, transformation, materials, and so on. From this scene, you can extract cameras, lights, transforms, and meshes into Nuke. This way, you can, for example, create a camera in Maya, export it in an .fbx file, and use the same camera again in Nuke.

To extract cameras, lights, transforms, and meshes into Nuke from .fbx files created in other applications, use the following nodes:

- for cameras, the Camera node
- for lights, the Light node
- for transforms, the Axis node
- for meshes (or NURBS curves/patch surfaces converted to meshes), the ReadGeo node.

All these nodes include similar controls for handling .fbx files, as you will notice from their descriptions below.

Note

The FBX SDK reads .fbx files produced by Autodesk MotionBuilder versions 5.5 and later. The FBX SDK writes .fbx files compatible with MotionBuilder (version 5.5 and later) and earlier versions of the Autodesk FBX SDK (6.0, 7.0, 2005.12, and later).

Tip

If you have trouble with FBX files, it may be because they were written with an older version of FBX. If they load very slowly, it is also possible that they are ASCII rather than binary. To get around these problems, you can use the FBX converter on the Autodesk web site. It converts between various different formats, including older FBX versions, ASCII, and binary, and is available on Windows, Mac OS X, and Linux.

To download the FBX converter:

1. Go to http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=10775855 .
2. Scroll down to **FBX Converter** and click on one of the links to start the download.

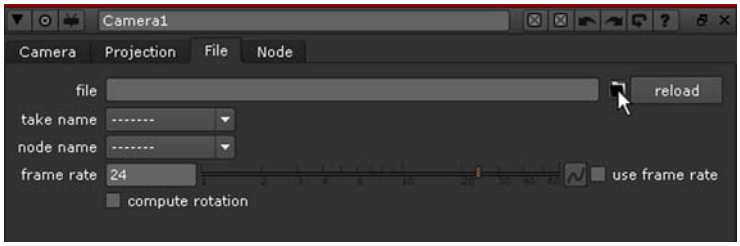
Importing Cameras from an .fbx File

The Camera node lets you read in the standard fbx cameras (Producer Perspective, Producer Top, Producer Bottom, Producer Right, Producer Left, Producer Front, Producer Back) and any other cameras.

Using one Camera node, you can only import one camera from an .fbx file. If you need to import several cameras, you need to use one Camera node per camera.

To import a camera from an .fbx file:

1. Select **3D** > **Camera** to insert a Camera node in the place where you want to add the camera in your script.
2. In the Camera controls, check **read from file**. When this is checked, the controls on the **File** tab are enabled, and you can use them to read in a camera from an .fbx file. Any controls whose values are read in from the .fbx file are disabled. You can still view these values and use them in expressions but, as long as **read from file** is checked, you cannot modify them. Modifying the values in the .fbx file, however, will affect the disabled values in the Camera controls, because these are reloaded from the .fbx file every time the node is instantiated.
3. To read in a camera from an .fbx file, click the folder icon on the **File** tab. Navigate to the .fbx file and select **Open**.



4. From the **take name** pulldown menu, choose the take you want to use from the .fbx file. Fbx files support multiple takes in one file. Usually, one of the takes is a default take with no animation.
5. From the **node name** pulldown menu, select the camera node you want to import from the .fbx file.
6. In the **frame rate** field, define a frame rate (frames per second) to sample the animation curves. To use this rate rather than the one defined in the .fbx file, check **use frame rate**.

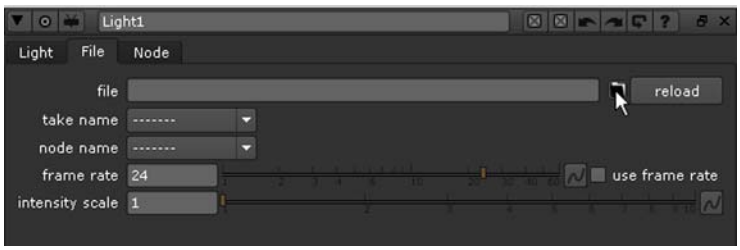
7. To have the camera rotation values calculated using the look up vector and look at position, check **compute rotation**. If you don't check this, Nuke uses the rotation channel from the .fbx file instead of computing a new one. The rotation values are always computed when there is a look at target.
8. If you want to modify the camera properties imported from the .fbx file, uncheck **read from file** on the **Camera** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
9. To reload the camera properties from the .fbx file, make sure **read from file** is checked and click the **reload** button on the **File** tab.

Importing Lights from an .fbx File

You can use the Light node to read in directional, point, and spot lights from .fbx scene files (for more information on these three light types, refer to *Lighting* on page 309). One Light node only reads in one light. Therefore, if your .fbx file contains three lights and you want to import all of them into Nuke, you need to use three Light nodes.

To import a light from an .fbx file:

1. Select **3D** > **Lights** > **Light** to insert a Light node in the place where you want to add the light in your script.
2. In the Light controls, check **read from file**. This enables the controls on the **File** tab, allowing you to read in lights from an .fbx file. It also disables all controls whose values will be filled from the .fbx file. You can still view these values and use them in expressions, but you cannot modify them, because they are read from the .fbx file. Any changes you make in the .fbx file will be reflected in these values of the Light node.
3. On the **File** tab, click the folder icon and browse to the .fbx file that contains the light you want to use. Click **Open**.



4. From the **take name** pulldown menu, select the take you want to use from the .fbx file. Fbx files support multiple takes in the same file. One of the takes is usually a default take without any animation.
5. From the **node name** pulldown menu, select the light node you want to import from the .fbx file.
6. If you want to override the frame rate used in the .fbx file to sample the animation curves, enter a new frame rate (frames per second) in the **frame rate** field. Check **use frame rate** to use the rate you entered (rather than the one in the .fbx file).
7. To scale the intensity channel values read from the .fbx file, adjust the **intensity scale** slider. If the light is too dark, increase this value.
8. If you want to modify the light properties imported from the .fbx file, uncheck **read from file** on the **Light** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.

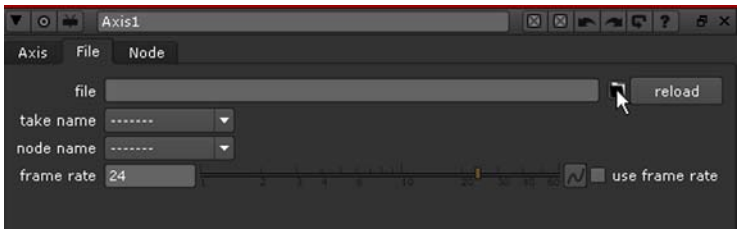
9. To reload the light properties from the .fbx file, make sure **read from file** is checked and click the **reload** button on the **File** tab.

Importing Transforms from an .fbx File

The Axis node reads in transforms, markers and nulls (locators) from .fbx files. You can use it to import one transform, marker, or null per Axis node.

To import a transform from an .fbx file:

1. Select **3D** > **Axis** to insert an Axis node in your script. Connect the Axis node to a Scene node.
2. In the Axis controls, check **read from file**. This enables the controls on the **File** tab, allowing you to import transforms from an .fbx file. It also disables controls whose values are filled in from the .fbx file. As long as **read from file** is checked, you cannot modify these values. You can, however, view them and use them in expressions. The values are reloaded from the .fbx file every time the node is instantiated, so any changes you make in the .fbx file's values will be reflected in the Axis controls.



3. On the **File** tab, click the folder icon to open the File Browser. Navigate to the .fbx file that contains the transform you want to use. Click **Open**.
4. From the **take name** pulldown menu, select the take you want to use from the .fbx file. Fbx files support multiple takes, one of which is usually a default take with no animation.
5. From the **node name** pulldown menu, choose the transform, marker, or null you want to import from the .fbx file.
6. If you do not want to use the frame rate from the .fbx file for sampling the animation curves, in the **frame rate** field, enter a new value (frames per second). To override the frame rate defined in the .fbx file and use the one you defined here, check **use frame rate**.
7. If you want to modify the transform properties imported from the .fbx file, uncheck **read from file** on the **Axis** tab and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
8. To reload the transform properties from the .fbx file, make sure **read from file** is checked and click the **reload** button on the **File** tab.

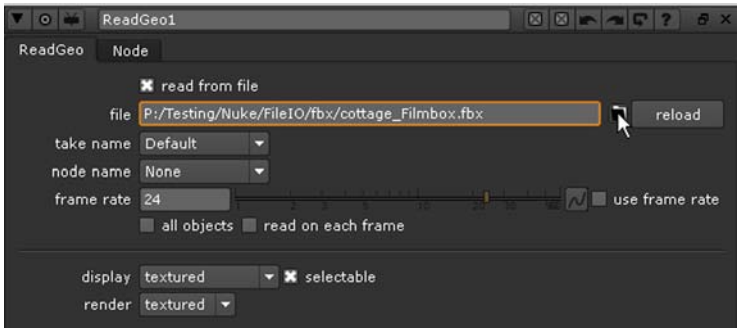
Importing Meshes from .fbx Files

The ReadGeo node lets you import meshes (or NURBS curves/patch surfaces converted to meshes) from .fbx files. Using one ReadGeo node, you can read in a single mesh or all the meshes in an .fbx file.

The mesh's vertices, normals, UV's, and vertex colours are read on a per frame basis or at frame 0. If there are any shape or cluster deformers, they are applied to the vertices. Materials or textures are not read in.

To import a mesh from an .fbx file:

1. Select **3D** > **Geometry** > **ReadGeo** to insert a ReadGeo node into your script.
2. In the ReadGeo controls, click the folder icon next to the **file** field and navigate to the .fbx file that contains the mesh you want to import. Click **Open**.



3. Make sure **read from file** is checked. This enables the file controls below. It also disables any controls whose values will be filled in from the .fbx file. You can view these values and use them in expressions, but as long as **read from file** is checked, you cannot modify them. Any changes in the .fbx file's values are reflected in the ReadGeo controls, however, because the values are reloaded from the .fbx file every time the node is instantiated.
4. From the **take name** pulldown menu, choose the take you want to use. Fbx files support multiple takes. Usually, one of them is a default take that contains no animation.
5. From the **node name** pulldown menu, select the mesh you want to import from the .fbx file.
6. To adjust the frame rate used to sample the animation curves, enter a new value (frames per second) in the **frame rate** field. The frame rate you enter is only used if you check **use frame rate**. Otherwise, the frame rate from the .fbx file is used.
7. If you want to import all the meshes in the .fbx file rather than just one, check **all objects**. This overrides whatever you have selected under **node name**.

   Each mesh you read in can have a transform. These are baked into the mesh points and imported into Nuke.
8. To reload the mesh on each frame, check **read on each frame**. You should do this when you are reading in all objects and the objects are animated.
9. If you want to modify the transform properties imported from the .fbx file, uncheck **read from file** and make the necessary modifications. As long as **read from file** is unchecked, your changes are kept.
10. If the meshes you are importing from the .fbx file are animated, check **read on each frame**. This reads in the meshes on each frame, preserving the animation.
11. To reload the transform properties from the .fbx file, make sure **read from file** is checked and click the **reload** button.

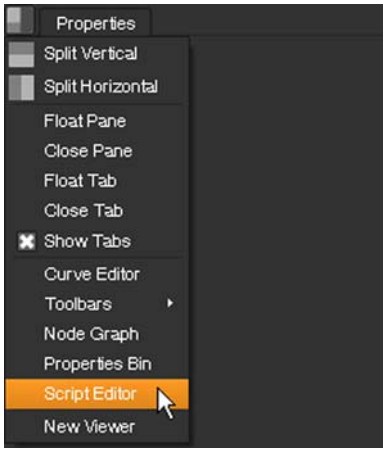
Importing Cameras from Boujou

Nuke is shipped with a script called import_boujou.tcl, which lets you load in cameras created with Boujou.

To import a camera from Boujou:

1. Save the Boujou camera solve as a .txt file.

2. In Nuke, click on a content menu button and select Script Editor. The Script Editor opens in the pane whose content menu you used.



3. In the input pane of the Script Editor (that is, the lower pane), enter **nuke.tcl("import_boujou")**. Click the **Run the current script** button on the top of the Editor, or press **Ctrl+Return** (**Cmd+Return** on a Mac).

4. In the File Browser that opens, navigate to the .txt file you saved in step 1.

A Camera, a ScanlineRender, and a Group node are loaded into Nuke. The Group node contains cylinders to represent points from Boujou.

Tip

You can also open Nuke's Boujou Text File Browser by doing the following:

1. Press **x** on the Node Graph to open the Nuke script command dialog.
2. In the dialog, check **TCL** (if it's not already checked).
3. In the command field, enter **import_boujou**.
4. Click **OK**.

These steps can be used to replace the first three steps in the above instructions.

Rendering a 3D Scene

The 3D Viewer displays the scene using an OpenGL hardware render. When you build a scene, Nuke renders high-quality output from the perspective of the camera connected to the ScanlineRender node. The rendered 2D image is then passed along to the next node in the compositing tree, and you can use the result as an input to other nodes in the script.

To render out a scene:

1. Make sure the rendering camera is connected to the ScanlineRender node.
2. Toggle the Viewer back to 2D.
3. Connect the output of the ScanlineRender node to the appropriate 2D nodes in your script.

Adjusting the Render Parameters

You can affect the rendered output by adjusting the various controls of the ScanlineRender node.

You can, for example, select the projection mode to do have different renderings of the scene, or change the global ambient colour. The global ambient colour is the overall colour added to the areas that are not illuminated. Without this colour, these areas appear black.

To select the projection mode:

From the **projection mode** pulldown menu, select:

- **render camera** to use the projection type of the render camera. This option is selected by default.
- **perspective** to have the camera's focal length and aperture define the illusion of depth for the objects in front of the camera.
- **orthographic** to use orthographic projection (projection onto the projection plane using parallel rays).
- **uv** to have every object render its UV space into the output format. You can use this option to cook out texture maps.
- **spherical** to have the entire 360-degree world rendered as a spherical map.

To change the global ambient colour:

Drag the **ambient** slider, or enter a value between 0 (black) and 1 (white) in the input field.

14 Working with Stereoscopic Projects

The title of this chapter is slightly misleading, as Nuke isn't actually limited to stereoscopic views, but rather provides multi-view support for as many views as you need. The views do not have to be stereo pairs, but since that is the most obvious application, this chapter mainly deals with stereoscopic projects.

In many ways, Nuke lets you work on stereoscopic material just like you would on any other images. However, there are also a few stereo-specific settings and nodes that you need to be aware of when compositing stereoscopic material. This following teaches you how to set up your stereo project, read in and view your images, use the stereo nodes, and render the final output.

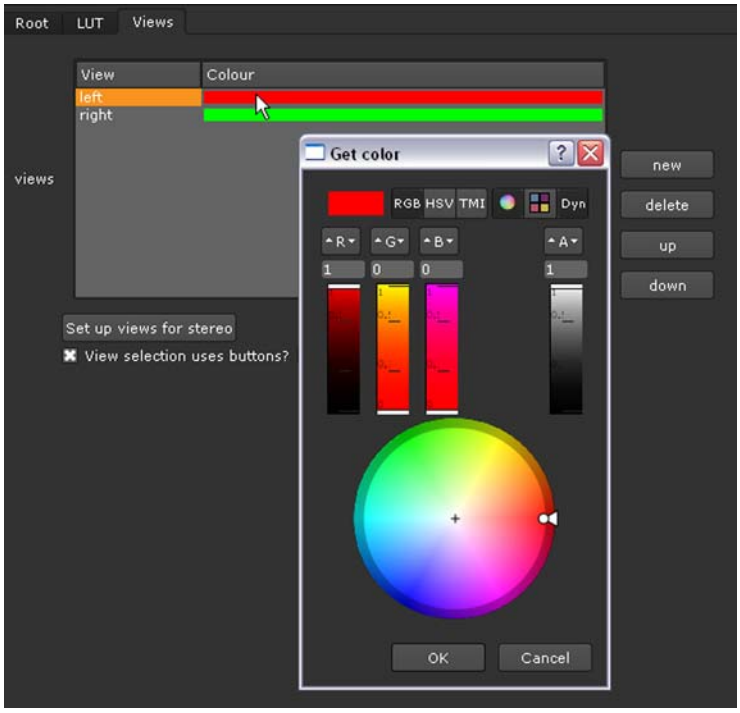
Setting Up Views for the Script

Before you start working on stereoscopic images, you need to set different views for the right and the left eye in the project settings. This allows you to process the individual views separately or both views together, and see the effect of your changes on each view.

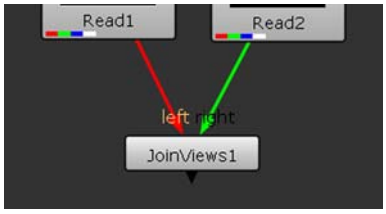
To set up views for your project:

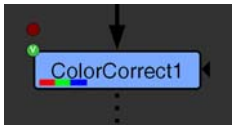
1. Select **Edit** > **Project settings**.
2. Go to the **Views** tab. The available views are listed in the **views** field.

3. If you want to remove the view called **main** and add views called **left** and **right**, click the **Set up views for stereo** button. The two views are assigned colours. To change the colours, double-click on the colour field and choose another colour from the colour picker that opens.

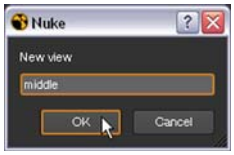


If you check **Use colours in UI?**, these colours are used in Node Graph connections, split views indicators on nodes, and Viewer and ShuffleViews node controls to make it easier to differentiate between views.

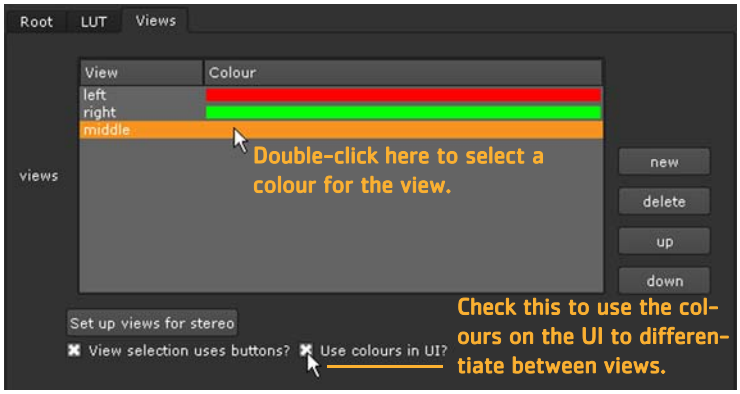




4. If you want to add other new views, click the **new** button.
5. In the dialog that opens, enter a name for the view, for example **middle**. Click **OK**.



6. Repeat steps 4 and 5 as necessary until you've got the views you want. You can assign colours to all views by double-clicking the area on the right of the view name.



7. To delete an unnecessary view, select the view from the list and click the **delete** button. Note that deleting a view does not remove references to it from the script, and any nodes that refer to the deleted view will produce an error.

You can now access the views in your project from the **view** pulldown menu of certain nodes' controls. You'll also notice that each view has its own button in the Viewer controls.

If you created many views, you may want them to appear in a pulldown menu rather than as separate buttons in the Viewer and certain node controls. To switch to using pulldown menus, uncheck **View selection uses buttons?** on the **Views** tab of the Project settings properties panel.

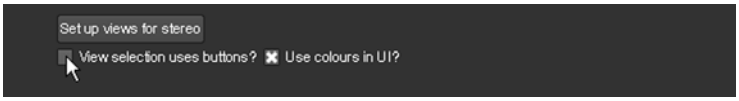


Figure 14-1: Setting the way views are selected in the Viewer.

If you are likely to need the same views in several projects, you may want to save the views you created in a template.nk script file. For more information on how to do this, see *Template Scripts* on page 431.

Loading Multi-View Images

Once you have set up the views, you are ready to read your images into Nuke. To make things easier, the images you read in should have the view name or the first letter of the view name in the filename, for example **filename.left.0001.exr**, **filename.l.exr**, or **lefteye-filename.0001.cin**.

If you are using .exr files, your files can contain both the input for the left eye and the input for the right eye, as .exr files support multiple views in a single file. With any other file types, you need to have separate files for the left and right inputs.

To read images in:

1. Select **Image** > **Read.**
2. Navigate to the files containing the images intended for either the left or right eye (or in the case of exr images, both eyes), and select **Open**.

3. Do one of the following:
   - If the images you want to read in contain a view name or the initial letter of one (for example, **left**, **right**, **l** or **r**) in their filenames, replace this with the variable **%V** or **%v** in the **file** field of the Read node's controls. Use **%V** to replace an entire view name (for example, **left** or **right**), and **%v** to replace an initial letter (for example, **l** or **r**). When a variable is used, Nuke reads in the missing inputs and combines all inputs into a single output.

     For example, if you read in **image.left.cin** and changed the name to **image.%V.cin**, Nuke would read in both **image.left.cin** and **image.right.cin** with the same Read node, provided that views called **left** and **right** existed in your project settings. Both input images would be combined into a single output.

     You can also use the **%V** and **%v** variables at a directory level. For example, let's say you have set up views called **testleft**, **testmiddle** and **testright**, and you have the following directories and files:

     mydirectory/testleft/image.testleft.cin

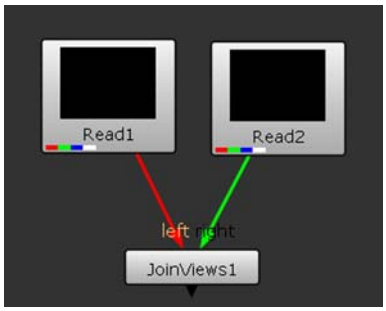
     mydirectory/testmiddle/image.testmiddle.cin

     mydirectory/testright/image.testright.cin.

     If you now read in image.testleft.cin and changed the pathname to **mydirectory/%V/image.%V.cin**, all three inputs would be read in with the same Read node.
   - If the images you want to read in do NOT contain a view name or the initial letter of one (for example, **left**, **right**, **l** or **r**) in the filenames and are not stereo .exr files, insert a Read node for each input and combine them into a single output using the JoinViews node (see below for instructions on how to do that).
   - If the images you want to read in are in the stereo .exr file format, you do not need to do anything. However, remember that not all .exr files are stereo .exrs. If you are using ones that are not, follow the instructions in the first two points.

To combine different views into a single output when the views are not indicated in the filenames:

1. Select **Image** > **Read** to read in your image sequences containing the different views.
2. To insert a JoinViews node, select **Views** > **JoinViews**.
3. Connect the inputs of the JoinViews node into the appropriate Read nodes. There should be an input for each view you have created in the project settings. The inputs are labeled with the name of the view.



If you have assigned colours to the views and checked **Use colours in UI?** on the **Views** tab of your project settings, the connecting arrows will reflect the view colours. If this does not happen and the arrows are black, you may have connected the inputs the wrong way around. Check that you have connected each Read node to the correct input of the JoinViews node.

Nuke combines the inputs into a single output.

Displaying Views in the Viewer

You can only display the views that exist in your project settings. To see a list of these views or add or delete views, select **Edit** > **Project settings** and go to the **Views** tab. For more information, see *Setting Up Views for the Script* on page 329.

To display a particular view:

1. Add a Viewer into your script if you haven't already done so.
2. On top of the Viewer controls, do one of the following:
   - If you have checked **View selection uses buttons?** in the project settings, click the button of the view you want to display. For example, click the **right** button (assuming you have a view called **right** in your script).

   - If you haven't checked **View selection uses buttons?** in the project settings, select the view you want to display from the pulldown menu.

Tip

You can also press the ; (semicolon) and ` (forward single quote) keys to move between different views in the Viewer.

To display two views next to each other:

1. Add a Viewer into your script if you haven't already done so.
2. If necessary, combine your views into a single output using the JoinViews node. For more information on how to do this, see page 332.
3. Select **Views** > **Stereo** > **SideBySide** to insert a SideBySide node in an appropriate place in your script.
4. In the SideBySide node's controls, select the two views you want to display from the **view1** and **view2** pulldown menus. View1 will be displayed on the left and view2 on the right.
5. If you want to display one view on top of another rather than next to it, check **vertical**. View1 will be displayed above view2.

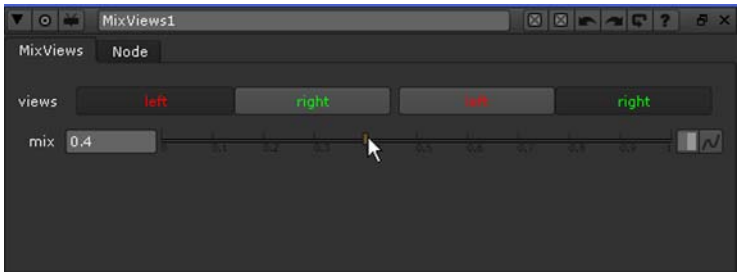
6. If you want to swap the views around in the Viewer, click the **swap** button.

The Viewer displays the two selected views simultaneously, so you can easily compare them.

To display a blend between two views:

1. Add a Viewer into your script if you haven't already done so.
2. If necessary, combine your views into a single output using the JoinViews node. For more information on how to do this, see page 332.
3. Select **Views** > **Stereo** > **MixViews** to insert a MixViews node into your script. This node displays a blend between two views in the Viewer, allowing you to check how elements in these views are aligned.

   Note that the MixViews node does not have an effect on the final output. It simply changes the way the images are displayed in the Viewer.
4. In the MixViews controls, use the **views** buttons or pulldown menus to select the two views to blend between.
5. To control the blend between the views, adjust the **mix** slider. Setting the slider to 0 or 1 displays only one of the views. Values between 0 and 1 produce different blends between the views.



Selecting which Views to Apply Changes to

By default, Nuke applies any changes you make to all views of the processed node. To apply changes to a particular view only (for example, the left view but not the right), you must first do one of the following:

- In the case of most nodes, split the view off in the node's controls.
- In the case of Paint nodes, select the view you want to process from the **view** pulldown menu in the node's controls.

These methods are useful, for example, when you want to perform the same operation on both views but use different values for each.

Splitting Views Off

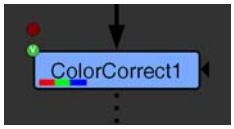
To split a view off:

1. Insert a process node (for example, ColorCorrect) in the appropriate place in your script.
2. If you haven't already done so, attach a Viewer to the node. From the Viewer's controls, select the view you want to make changes to.

3. Open the node's controls.
4. Click the view button next to the control you want to adjust. From the menu that opens, select **Split off [view name]**. For example, to apply changes to a view called **left**, select **Split off left**. You can also split all the node's controls by selecting **Split all knobs** from the right-click menu.

An eye appears on the view button and the node gets a small green dot on it in the Node Graph to indicate that views have been split off.

If you have assigned colours to the views and checked **Use colours in UI?** in your project settings, dots also appear on the node to indicate which views have been split off. For example, if you are using red for the left view and split off that view, a red dot appears on the node.



Any changes you now make using the control in question are only applied to the view you chose to split off. Changes to controls that have not been split off are still applied to all views.

To show separate values for each view:

Once you have split off a view, you can apply changes to the existing views separately. Simply click on the small arrow on the left side of a control you have split off. This divides the control so that you can give separate values for each view.

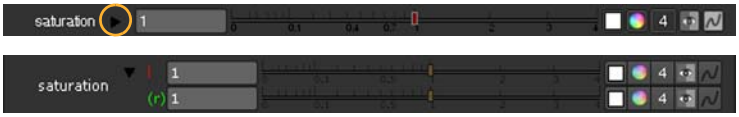


Figure 14-2: Adjusting a split control for only the split view and for all views separately.

To unsplit views:

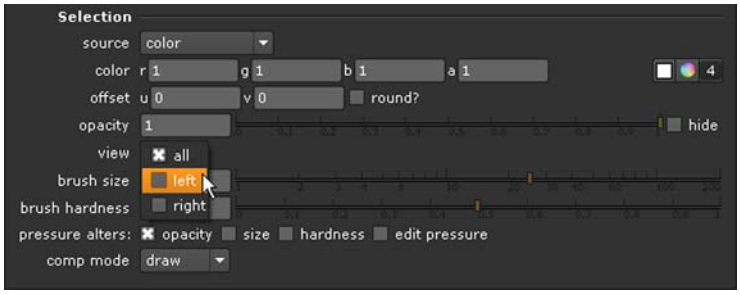
1. In the node's controls, click the view button.
2. From the menu that opens, select **Unsplit [view]**. For example, to unsplit a view called **left**, you'd select **Unsplit left**.
3. Repeat step 2 for all views you want to unsplit.

The view is unsplit, and all changes you made after splitting it off are lost.

Selecting the View to Process When Using the Paint Node

To select the view to process:

1. Open the Paint node's controls.
2. From the **view** pulldown menu, select the view you want to process. To apply changes to all views at the same time, select **all**.



3. If you selected to process just one view, make sure you are viewing the selected view in the Viewer when making your changes.

Performing Different Actions on Different Views

In case you need to perform totally different actions on the two views, you can add a OneView node to separate one view for processing.

To extract a view for processing:

1. Select **Views** > **OneView** to insert a OneView node in an appropriate place in your script.
2. In the OneView node's controls, select the view you want to make changes to from the **view** pulldown menu.

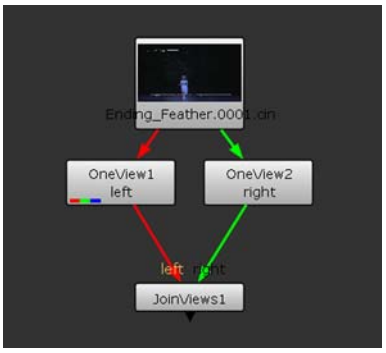
All views are extracted, and any changes you make are only applied to the view you selected (regardless of which view you are displaying in the Viewer).

To make changes to a different view, select it from the OneView node's **view** pulldown menu.

To merge views from two separate streams, select **Views** > **JoinViews** to combine the views (or delete the OneView node from your script).

If you need to extract all views, process them individually, and then merge them together, use the **Split and Join** menu item. This menu item is actually a combination of the OneView and JoinViews nodes. It first extracts all the views you have set up in your project settings and then merges them back together. It's no different to using several OneView nodes together with a JoinViews node, but makes working faster, because you do not need to add each node in a separate go. To use the menu item, select **Views** > **Split and Join**.

For example, if you have created views called **left** and **right** in your project settings and use a **Split and Join** menu item after your Read node, you get the following node tree:



You can then add any necessary nodes, such as colour corrections, between the OneView and JoinViews nodes.

Reproducing Changes Made to One View

When rotoscoping, creating paint effects, or doing other operations dependent on image locality, you can have changes made to one view automatically reproduced in the other. This applies to the Paint node, the Bezier node, and any nodes, groups, or gizmos that have controls for x and y coordinates.

To reproduce changes made with the above nodes, groups, or gizmos, you need a disparity field that maps the location of a pixel in one view to the location of its corresponding pixel in

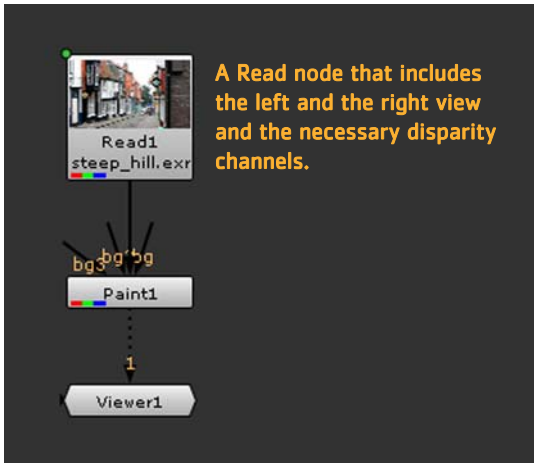
the other view. You can create a disparity field using The Foundry's O_DisparityGenerator plug-in, which is included in the Ocula plug-in set, or a 3D application. Once you have the disparity field, you can store it in the channels of an .exr file or use the ShuffleCopy node to add the disparity channels in the data stream where you need them.

If you have Ocula installed, you can choose between reproducing your changes using the disparity field or Ocula. If you select to use Ocula, extra refinements are done when correlating the changes from one view to the other. This can improve the results when working with live action footage. When working with CG images, however, the disparity maps should be accurate to begin with and produce good results even without Ocula.

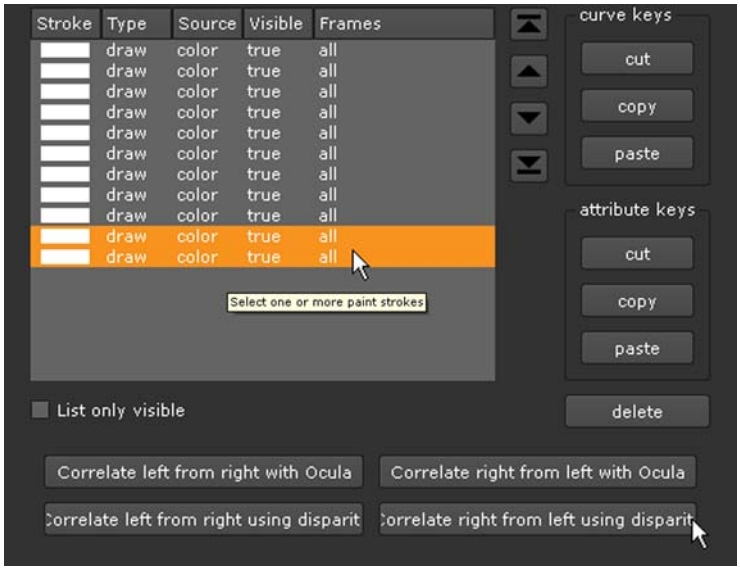
Reproducing Paint Strokes

To create a paint stroke on one view and have it automatically generated for the other:

1. Make sure there is a disparity field upstream from the image sequence you want to paint on. If the image sequence is an .exr file, the disparity field can be included in its channels. Otherwise, you can use a ShuffleCopy node or Ocula's O_DisparityGenerator plug-in to add it in the data stream.
2. Insert a Paint node after the image sequence you want to paint on.



3. In the Paint node controls, select **all** from the **view** pulldown menu. Display the view you want to paint on in the Viewer.
4. With the Paint controls open, draw a paint stroke in the Viewer.
5. Select the paint stroke in the Paint node controls.
6. Click **Correlate [view] from [view] using disparity** . For example, if your paint stroke was in the correct position in the left view but not the right, you'd select **Correlate right from left using disparity**.



This adds the disparity vectors in the map to the current values, creating the corresponding paint stroke for the other view.

If you have The Foundry's Ocula plug-ins installed, you can also click **Correlate [view] from [view] with Ocula**, for example **Correlate left from right with Ocula**. This way, extra refinements are made when creating the corresponding paint stroke, and the results may be more accurate.

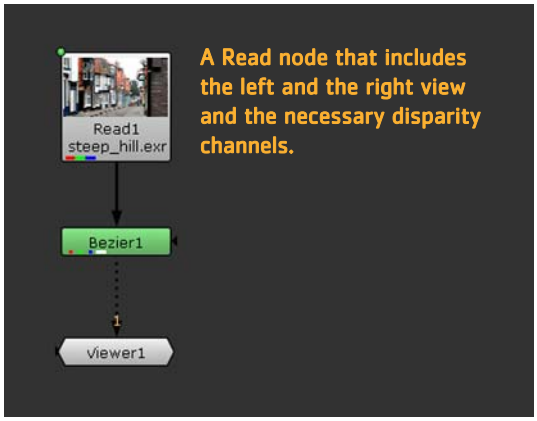
With Ocula installed, you can also use the **Ocula settings** controls (**blocksize** and **searchsize**) on the Paint node.

7. In the Viewer, switch between the two views to compare the original and the correlated paint strokes.
8. If you want to adjust the paint stroke further, you need to adjust both views independently. Adjustments you make to one view are not automatically generated for the other.

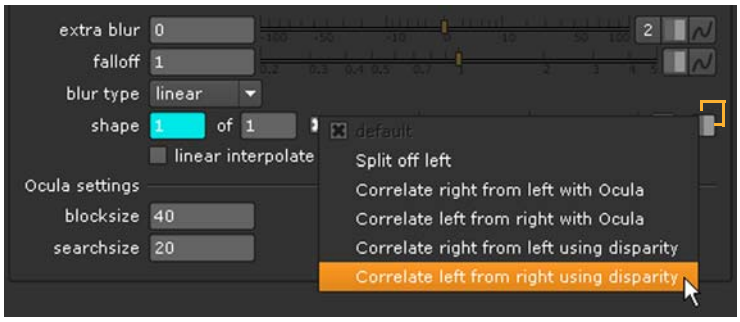
Reproducing Roto Shapes

To create a roto shape on one view and have it automatically generated for the other:

1. Make sure there is a disparity field upstream from the image sequence you are manipulating. If the image sequence is an .exr file, the disparity field can be included in its channels. Otherwise, you can use a ShuffleCopy node or Ocula's O_DisparityGenerator plug-in to add it in the data stream.
2. Insert a Bezier node after the image sequence you are manipulating.



3. With the Bezier controls open, create a roto shape for one view.
4. Go to the Bezier controls. From the View menu next to the **shape** control, select **Correlate [view] from [view] using disparity**, for example **Correlate left from right using disparity**. This adds a corresponding Bezier shape to the other view.



If you have The Foundry's Ocula plug-ins installed, you can also select **Correlate [view] from [view] with Ocula**. This way, extra refinements are made when creating the corresponding paint stroke, and the results may be more accurate.

With Ocula installed, you can also use the **Ocula settings** controls (**blocksize** and **searchsize**) on the Bezier node.

5. If you want to adjust the Bezier shape further, you need to adjust both views independently. Adjustments you make to one view are not automatically generated for the other.

Reproducing X and Y Values

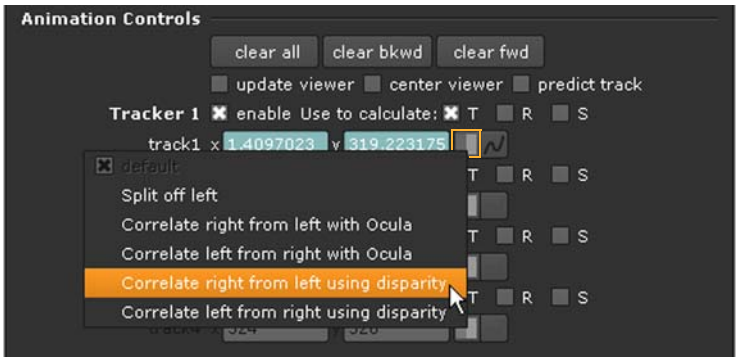
Whenever there are values in any x or y control in Nuke for one view, you can automatically generate the corresponding values for the other view. This is true for both nodes and gizmos. For example, you can use a Tracker node to track something in one view, and then have the track's x and y position generated for the other view automatically.

To produce x and y values for one view and have them automatically generated for the other:

1. Make sure there is a disparity field upstream from the image sequence you are manipulating. If the image sequence is an .exr file, the disparity field can be included in its channels.

Otherwise, you can use a ShuffleCopy node or Ocula's O_DisparityGenerator plug-in to add it in the data stream.

2. Insert a node that has an x and y control after the image sequence you are manipulating.
3. Attach a Viewer to the node you added in the previous step, and make your changes in one view.
4. From the View menu next to the **x** and **y** controls, select **Correlate [view] from [view] using disparity**, for example **Correlate right from left using disparity**. This generates the corresponding x and y values for the other view.



If you have The Foundry's Ocula plug-ins installed, you can also select **Correlate [view] from [view] with Ocula**. This way, extra refinements are made when creating the corresponding x and y values, and the results may be more accurate.

5. If you want to adjust the x and y values further, you need to adjust both views independently. Adjustments you make to one view are not automatically generated for the other.

Swapping Views

You can rearrange the views in your script using the ShuffleViews node. For example, you can swap the left and right views around in the pipeline, so that Nuke uses the left input for the right eye and vice versa.

To rearrange views:

1. Select **Views** > **ShuffleViews** to insert a ShuffleViews node in an appropriate place in your script.
2. In the ShuffleViews controls, click **add** as necessary.
3. Use the buttons or pulldown menus to select which view to replace with which. For example, to swap the left and right views around, you need to make the following selections:
   - On one row, select **left** under **get**, and **right** under **from** ("get left from right"). The left view is now replaced with the right view.
   - On another row, select **right** under **get**, and **left** under **from** ("get right from left"). The right view is replaced with the left view.

If there aren't enough rows of buttons or pulldown menus on the ShuffleViews node's properties panel, click the **add** button to add a row.

To remove unnecessary rows in the ShuffleViews node's controls, click the **delete** button next to the row you want to remove.

Converting Images into Anaglyph

You can use the Anaglyph node to convert your inputs into anaglyph images, which produce a 3D effect when viewed with 2-colour anaglyph glasses.

To convert your images into anaglyph:

1. Select **Views** > **Stereo** > **Anaglyph** to insert an Anaglyph node in an appropriate place in your script.
2. Use the **views** controls in the Anaglyph properties panel to select which views you want to use for the left and the right eye.

   Nuke converts the input images into greyscale anaglyph images. The left input is filtered to remove blue and green, and the right view to remove red.

3. To add colour into the images, drag right on the **amtcolour** slider, or insert a value between 0 (greyscale) and 1 (coloured) into the **amtcolour** input field.

If the images include areas that are very red, green, or blue, adding more colour into them may not produce the best possible results.

4. To invert the colours and use the red channel from the right input and the blue and green channels from the left, check the **(right=red)** box.

5. To control where the images appear in relation to the screen when viewed with anaglyph glasses, enter a value in the **horizontal offset** input field. To have the images appear in front of the screen, you would usually enter a negative value. To have the images appear further away, you would usually enter a positive value. (This is not the case if you have swapped the left and right views around.)

Changing Convergence

The ReConverge node lets you shift *convergence* (the inward rotation of the eyes or cameras) so that any selected point in the image appears at screen depth when viewed with 3D glasses. This point is called the *convergence point.* It is the point where the lines of sight from the two cameras meet.

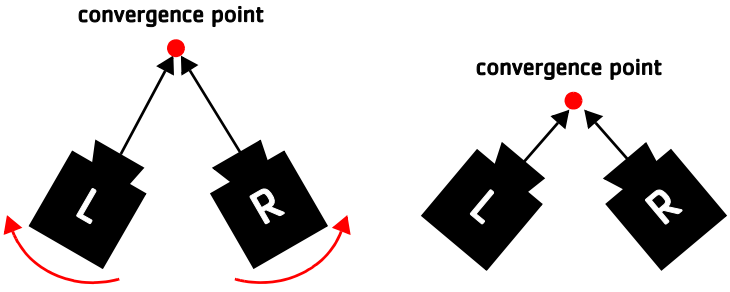


Figure 14-3: Changing convergence moves the point where the lines of sight from the two cameras meet.

At the convergence point, the different views in the image are aligned and appear at screen depth when viewed with 3D glasses. Anything behind the convergence point appears behind the screen, while anything in front of it seems to pop out of the screen. This is illustrated in Figure 14-4.

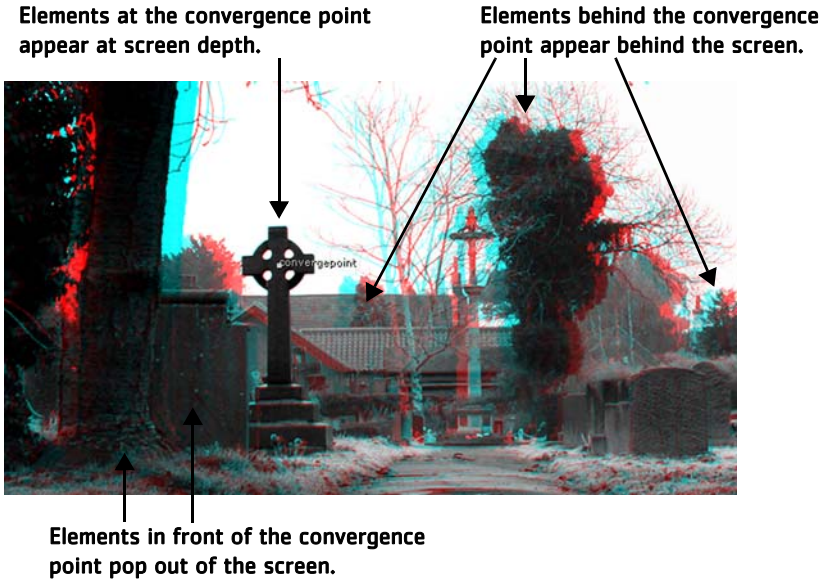

Elements at the convergence point appear at screen depth.
Elements behind the convergence point appear behind the screen.
convergepoint
Elements in front of the convergence point pop out of the screen.

Figure 14-4: Convergence controls where elements in the image appear in relation to the screen when viewed with 3D glasses.

Changing convergence changes the perceived depth of the images. It moves all the elements in the image backwards or forwards a fixed distance while keeping the distance between them the same. This is illustrated in Figure 14-5, where the grey rectangles represent elements depicted in a stereo image.

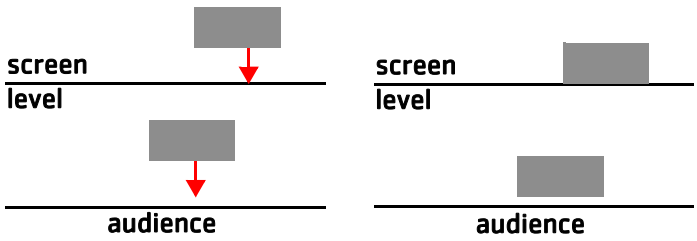


Figure 14-5: Changing convergence changes the perceived depth of the images.

Often, the element of an image that appears closest to the audience is used as the convergence point. However, to make an element in your image jump out of the screen, you need to converge on something behind this element.

To calculate the convergence shift, the ReConverge node needs a disparity field that maps the location of a pixel in one view to the location of its corresponding pixel in the other view. To create the disparity field, you can use The Foundry's O_DisparityGenerator plug-in, which is part of the Ocula plug-in set. Alternatively, you can create the disparity field in a 3D application. Once you have the disparity field, you can store it in the channels of an .exr file or use the ShuffleCopy node to add the disparity channels in the data stream where you need them.

If you have Ocula installed, you can choose to use it when changing convergence. If you select to use Ocula, extra refinements are done when performing the convergence shift. This can improve the results when working with live action footage. When working with CG images, however, the disparity fields should be accurate to begin with and produce good results even without Ocula.

Note that the ReConverge node only shifts views horizontally, not vertically.

To change the convergence point of a stereo image:

1. Make sure there is a disparity field upstream from the image sequence whose convergence you want to change. If the image sequence is an .exr file, the disparity field can be included in its channels. Otherwise, you can use a ShuffleCopy node or Ocula's O_DisparityGenerator plug-in to add it in the data stream.
2. From the Toolbar, select **Views** > **Stereo** > **ReConverge** to insert a ReConverge node after the image sequence whose convergence you want to adjust.
3. Attach a Viewer to the ReConverge node.
4. To better view the effect of the ReConverge node, insert an Anaglyph node (**Views** > **Stereo** > **Anaglyph**) between the ReConverge node and the Viewer.

5. If you have The Foundry's Ocula plug-ins installed, check **Use Ocula if available** In the ReConverge properties panel. This way, extra refinements are made when changing the convergence and the results may be more accurate.
6. Make sure the ReConverge properties panel is open. You should see the convergence point overlay in the Viewer. Drag the point on top of the point you want to appear at screen level when viewed with 3D glasses. The convergence shifts to this location.

   You can also move the convergence point by entering the point's x and y coordinates in the **Covergence upon** fields.
7. By default, the ReConverge node moves the right view to achieve the convergence shift. However, if you like, you can use the **Mode** pulldown menu in the ReConverge controls to move the left view instead (select **shift left**) or move both views equally (select **shift both**).

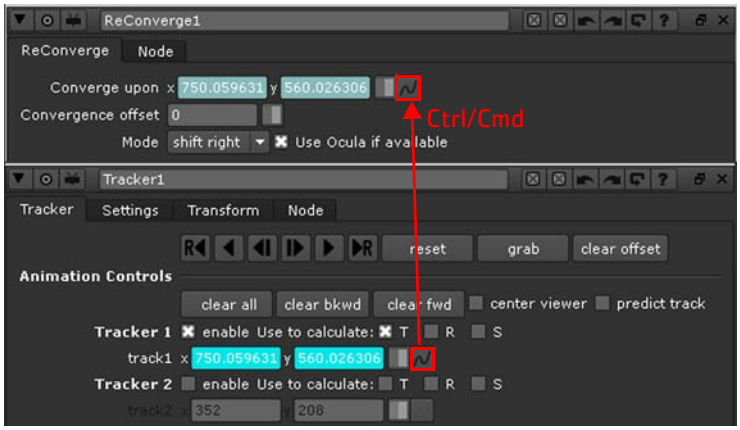
8. If necessary, adjust the offset for convergence (in pixels) in the ReConverge controls. To bring all elements of your image forward from the screen level, enter a positive value in the **Convergence offset** field. To move all elements further away, enter a negative value.

It is also possible to use the same element as the convergence point throughout the image sequence. You can, for example, have the same actor always appear at screen depth. To converge on the same element throughout the sequence, link the ReConverge node with a Tracker node.

To use the same element as the convergence point throughout the image sequence:

1. Insert a Tracker node after the image sequence whose convergence you want to adjust.

2. Track the point that you want to appear at screen level throughout the sequence. For more information on how to use the Tracker node, refer to Chapter 6: *Tracking and Stabilising* on page 175.
3. When you have the track animation data, apply it to the O_ReConverge node's **Converge upon** control via a linking expression. The easiest way to do this is to **Ctrl**/**Cmd**+drag the animation button next to the **track1** controls on top of the animation button next to the **Convergence upon** control.



Rendering Stereoscopic Images

You can render several views using a single Write node. When using the stereo extensions for the .exr file format, Nuke writes the output of both views into a single file. With any other file types, the views are written into their respective files.

To render .exr files:

1. Select **Image** > **Write** to insert a Write node in an appropriate place in your script.
2. In the Write node's controls, select **exr** from the **file type** pulldown menu.
3. From the **views** pulldown menu, select the view(s) you want to render, for example **left, right**.
4. Adjust any other Write controls as necessary and click **Render**. Nuke prompts you for the frames to render.

Nuke writes several views into a single file.

To render files that are not in the .exr file format:

1. Select **Image** > **Write** to insert a Write node in an appropriate place in your script.
2. In the Write nodes' controls, select the file type of your images from the **file type** pulldown menu.
3. When entering names for the rendered image sequences, you can use the variable **%V** (with a capital V) to represent the words **left** and **right** (or any other full view names) in the filenames, for example **filename.%V.%04d.exr**. To represent the letters **l** and **r** (or the first letters of any views), use the variable **%v** (with a lower-case v) instead. When rendering, Nuke then fills this in with left, right, l, or r, and renders all views you specify in the next step.
4. Adjust any other Write controls as necessary and click **Render**. Nuke prompts you for the frames to render as well as the views to execute (assuming you have set up several views in the project settings).

Nuke renders several views, but writes them into separate files. If you did not specify a view in the filenames (using either the name of the view, its first letter, or a variable), you can only render one view.

Note

For command line renders, you can pass the **-view** argument with a list of view names to render, separated by a comma. If you do not specify a **-view** argument, Nuke renders all views.

15 Previews and Rendering

Nuke supports a fast, high-quality internal renderer, with superior colour resolution and dynamic range without a slowdown in the workflow. These are some of the key features of Nuke's rendering engine:

- Multi-threaded rendering to take advantage of multiple processors in its calculations.
- Scanline (as opposed to buffer-based) rendering allows you to immediately see portions of render output.
- Calculations performed with 32-bit precision, using linear light levels.

This chapter teaches you how to use the renderer's various features to preview a script's output and generate its final elements.

Previewing Output

This section explains how to preview individual frames in a Nuke Viewer window, and also how to render a flipbook for a sequence of frames.

Previewing in a Nuke Viewer

When you connect a Viewer to a given node's output (by selecting the node and pressing a number key), Nuke immediately starts rendering the output in the Viewer using all available local processors.

Keep in mind the following tips in order to speed up this type of preview rendering:

- First, if you don't need to evaluate the whole image, zoom into the area of interest (see *Zooming* on page 46 for more information). Nuke will then render only the portion of scan lines visible within the Viewer.
- Alternatively, you can use the Viewer's region of interest (ROI) feature to render only a portion of the image, while seeing that result in the context of the whole image.

To enable the ROI render feature:

1. Press **Alt+W** over the Viewer. The Viewer's **ROI** button turns red, indicating that the feature is enabled.
2. Drag on the Viewer to draw the region of interest. The Viewer will now render only the pixels within the region.

To edit the position or size of current ROI:

1. Click the **ROI** button until it turns red. The overlay for the current ROI appears in the Viewer.
2. To reposition the ROI:

   Drag to select all sides of the ROI, then drag the ROI to the desired location.
3. To resize the ROI:
   - Drag any corner of the ROI until you achieve the desired size.

To disable the ROI render feature:

Click the Viewer's **ROI** button. It turns gray, signalling that it is off. The Viewer will now render all of the visible image.

Flipbooking Sequences

Flipbooking a sequence refers to rendering out range of images (typically at proxy resolution), then playing them back in order to accurately access the motion characteristics of added effects.

You have two options for flipbooking within Nuke:

- You can enable automatic disk caching of rendered frames, then play these frames back using Nuke's native Viewer. This option does *not* let you define a specific playback rate.
- Or you can render out a temporary image sequences to FrameCycler, a RAM-buffering playback utility which is automatically installed with your copy of Nuke and plays back sequences at the defined frame rate.

Flipbooking within Nuke

To flipbook image sequences inside the Nuke Viewer, you must first enable the automatic disk caching of rendered frames. You do so by setting two preferences that define the location and size of the cache.

Once you set these preferences, the Nuke Viewer automatically saves to disk full float versions of every frame it displays. When you play through sequences in the Viewer, it reads, where possible, from this cache of prerendered images, making real-time play back possible (depending, of course, on image resolution and your hardware configuration).

To enable automatic disk caching of rendered frames:

1. Click **Edit > Preferences** to display the Preferences dialog.
2. In the **disk cache** field, enter the pathname of the directory in which you want to store the flipbook images (for example, c:/temp).
3. Using the **disk cache size** control, select the number of gigabytes you want to allow the image cache to consume.
4. Click the **Save Prefs** button to update preferences and then restart Nuke.

The Viewer will now cache each frame it displays in the directory specified. When you click the playback buttons on the Viewer, or drag on the scrub bar, Nuke will read in images from this cache.

Note that the cached images have unique names reflecting their point of output location in the script. This means that you can cache images from multiple nodes in the script without overwriting previously cached images.

Flipbooking within FrameCycler

To flipbook an image sequence inside FrameCycler:

1. Select the node whose output you wish to see flipbooked.

Note

If you select a Write node in the step above, you must first click its **Render** button in order to manually render its output to the destination defined in the **file** field. This step is necessary only in the case of Write nodes.

2. Select **Render** > **Flipbook selected** (or press **Alt**+**F**).

   A dialog opens.

3. In the **Frames to flipbook** field, enter the first and the last frame you want to preview, separated by a comma (for example, **1,35**). Click **OK**.

   Nuke renders as a temporary sequence the output of the selected node using the frame range and resolution defined in the script's settings. This may take a few moments.

4. Once the render is complete, Nuke launches Framecycler and loads in the temporary sequence. You can play it back and view it using Framecycler's media controls.

Note

Framecycler comes packed with many features to complement flipbooking. You can, for example, attach sound files to the image sequence, cut it, or splice it together with other image sequences. Surf to **www.iridas.com** for more information.

Rendering Output

Nuke can render images locally— on your workstation— or it can be setup to render images on a network render farm. Before rendering, verify that your project settings have the correct output format and proxy format selected.

Render Resolution and Format

Before rendering a script, it's important to check what is the currently active mode: the full-size or the proxy mode. Nuke executes all renders at the currently active scale. Thus, when a script is rendered in proxy mode, processing will be done at the proxy scale and image output will go to the file name in the Write node's **proxy** field. If you do not specify a proxy file name, the render will fail with an error. It never resizes the proxy image, and it will not write the proxy image over the full-size one.

To view and change the proxy resolution for the current script file, choose **Edit > Project settings** from the menu bar, or press **S** with the mouse pointer over the Node Graph or the Properties Bin.

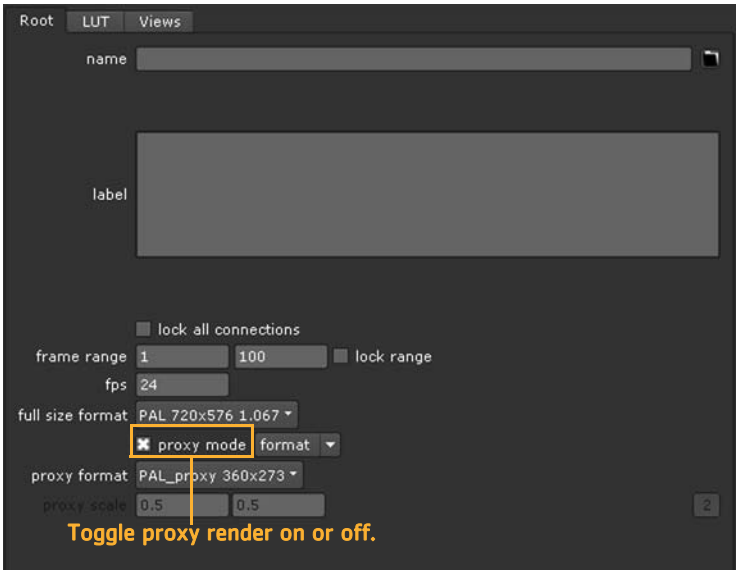


Figure 15-1: Changing the output resolution under Project settings.

From the Project settings properties panel, you can select a new render format from the list of predefined resolutions, and toggle proxy rendering. You can also choose the **new** option under either **full size format** or **proxy format** or use the **proxy scale** fields to define custom render resolutions for the composite. When rendering in proxy mode, use the pulldown menu on the right to select whether to use the resolution defined under **proxy format** or **proxy scale**. For more information on these settings, refer to *Proxy Format and Proxy Scale* on page 92.

Output (Write) Nodes

With the correct resolution and format selected, you then insert Write nodes to indicate where you want to render images from the script.

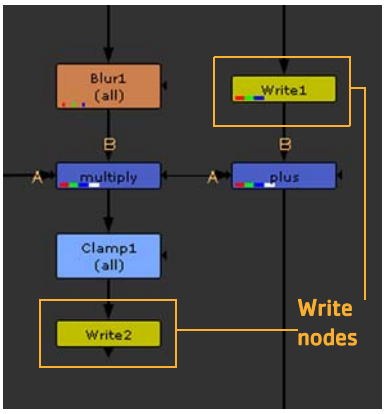


Figure 15-2: Inserting Write nodes for rendering.

One Write node is usually placed at the bottom of the compositing tree to render the final output. However, Write nodes have both input and output connectors, so they may be embedded anywhere in the compositing tree.

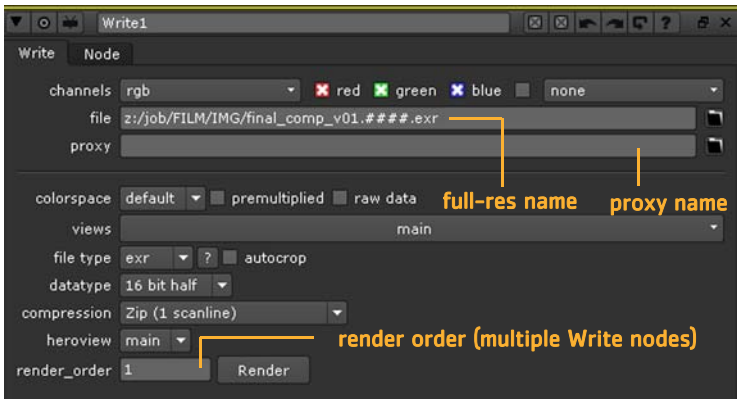
You can execute renders for a single Write node or all Write nodes in your compositing script.

To render a single Write node:

1. Select the node in the script from which you want to render an image.
2. Choose **Image > Write** (or press **W** over the Node Graph). Nuke attaches a Write node and opens its properties panel.
3. Connect a Viewer to the Write node you want to render and verify that the correct resolution is displayed for output. If necessary, press **Ctrl/Cmd+P** to toggle between full-res and proxy resolution. The displayed output resolution will be used for rendering.

.

4. In the properties panel, click the **file** or **proxy** field's folder icon (depending on whether you want to render high res or low res images) and browse to the directory where you want to store the rendered sequence. For instructions on using the file browser, see *Using the File Browser* on page 83.



5. After the path, type a name for the rendered image and then click **OK**. If you're rendering an image sequence, include the frame number variable (for example, **%04d**) in the name.

   See *To render selected or all Write nodes in the script:* below for examples of valid file names with the frame number variable.

6. If necessary, adjust the following controls:
    - Using the **channels** pulldown menu and checkboxes, select the channels you want to render.
    - From the **colourspace** pulldown menu, select which lookup table to use when converting between the images' colour space and Nuke's internal colour space.
    - From the **file type** pulldown menu, select the file format for the rendered images. If you don't specify a file format, Nuke uses the extension in the file name to figure out the format.
7. In the Write node properties, click the **Render** button.
8. Nuke prompts for a frame range. Enter the start and end frames, separated by a comma (for example, **1,100**), and then click **OK**

You can see the progress of your render in the status window that appears. When the render is complete, the rendered images are added to the directory you specified in step 4.

Tip

When rendering with the Write node, you can force a certain data type by adding the data type and a colon before the file path. For example, you can enter ftiff:C:\Temp\test.tif as the file path to render a file whose data type is ftiff and extension tif.

Tip

If you are rendering .mov files, you can choose the QuickTime codec from the **codec** pulldown menu, and adjust advanced codec options by clicking the **advanced** button.

Note

If you are using 64-bit Windows, you cannot render QuickTime files. This is because Apple has not released QuickTime for 64-bit Windows.

Note

When writing QuickTime files, some users have reported receiving the following error at the end of rendering:

Failed to flatten movie data: the movie is open in another application.

This is because the output file has been opened by some other process. This can be the Finder on Mac OS X trying to show a preview, an OS file system search trying to index the file, or a virus checker, for example. The workaround is to turn off **Fast Start** in the Write node controls to skip the flattening process that is affected.

To render selected or all Write nodes in the script:

1. Connect a Viewer to a Write node you want to render and verify that the correct resolution is displayed for output.
2. If necessary, press **Ctrl/Cmd+P** to toggle between full-res and proxy resolution. The displayed output resolution will be used for rendering.
3. Do one of the following:
   - With the desired Write node selected, choose **Render** > **Render selected** (or press **F7**).
   - Choose **Render** > **Render all** (or press **F5**).
4. Nuke prompts for a frame range. Enter the start and end frames, separated by a comma (i.e., **1,100**) and then click **OK**.

File Name Conventions for Rendered Images

There is no parameter in the Write node to specify output format. Instead, format is indicated by a prefix or an extension when you type the file name. Here is the appropriate syntax:

```
<prefix>:/<path>/<name>.<frame number variable>.<extension>
```

The optional *<prefix>:* can be any valid extension. *<path>* is the full pathname to the directory where you want to render. The *<frame number variable>* is usually entered as **%04d**, with the number 4 indicating frame numbers padded to four digits.

You can change the padding by substituting 4 with any single-digit number. For example, two-digit padding would be **%02d**, three-digit would be **%03d**, and five-digit would be **%05d**. With these conventions in mind, suppose you want to save an image sequence called "final_comp_v01" to the TIFF16 format. Here are examples of names that will work in the Write node:

```
tiff16:/<path>/final_comp_v01.%04d.tiff
/<path>/final_comp_v01.%04d.tiff16
tiff16:/<path>/final_comp_v01.%04d
```

All extensions supported by Nuke may be used for the prefix. See *Appendix B: Supported File Formats* on page 557 for a complete list of recognized extensions.

When a prefix is used, it takes precedence over the format represented by an extension. In fact, the prefix makes the extension unnecessary.

You could, for example, enter **exr:/<path>/%03d** as the file name, and this would create an OpenEXR image sequence with frame numbers only, padded to three digits.

Tip

You can also use **####** instead of **%04d**, **###** instead of **%03d**, **##** instead of **%02d**, and so on.

Render Farms

Nuke supports virtually all third-party and proprietary render-queuing software. By integrating Nuke with such a system, the render load can be distributed across all the Nuke-licensed machines on your network, whether Windows, Mac, or Linux-based.

Your installation of Nuke may be configured to send jobs to a network render farm, which is usually made available under the Render menu (i.e., **Render > Render**). However, because this option must be customized for each studio, you should check with your system administrator for instructions on how to send a Nuke script for network rendering.

16 Expressions

This chapter is intended as a primer on how to apply expressions (programmatic commands) to Nuke parameters. It explains how to perform some common tasks with expressions (for example, how to link the values of one parameter to another), and concludes with a table all the functions that you may include as part of an expression.

Understanding Expression Syntax

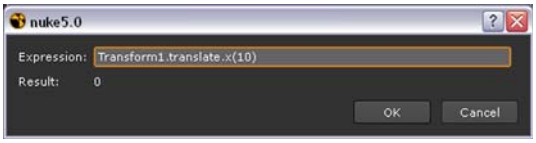
You enter Nuke expressions in the expression dialog, which you can open by pressing the equals sign (=) on a parameter. In the dialog, you can enter text that either references values from other parameters (creating a linking expression) and/or applies mathematical functions of some kind to the current values.

Linking Expressions

Through expressions, you can link the parameters from one node and control the values of the parameters in other nodes. When creating a linking expression, type the elements listed in the table below; remember to separate each element with a period.

Element	Description
Node name	The node with the source parameter (i.e., **Transform1**).
Parameter name	The name of the parameter with the source value (for example, **translate**).The name is defined internally, and may not match the parameter's label that appear in the Nuke interface. If necessary, hover over the parameter's field with your mouse pointer and its name will appear in the pop-up tool tip.
Child parameter name (optional)	Some parameters include child parameters, such as the fields for x and y axes, or red, green, and blue colour channels). Child parameter names do match the label that appears before the parameter's field (for example, **x**).
Time (optional)	By default, linking expressions pull values from the current frame number, but you can read values from other frames, either statically or dynamically (that is, with a temporal offset). If you want to read in a static value for a given frame, you just type that frame number inside a set of parenthesis (for example, **(10)**). If you want to read in dynamic values but with an offset in time, type **t**, the variable for time, followed by a **+** (for a forward offset) or **-** (for a backward offset), followed by a number representing the number of frames worth of offset. For example, typing **(t-2)** would capture values that are two frames back from the current frame.

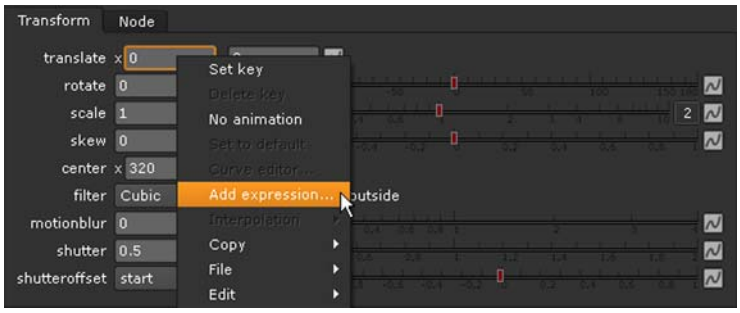
Thus, to create a linking expression that pulls the value from a Transform node's x translation field at the tenth frame, you would type **=** on a parameter to open the expression dialog, and then enter **Transform1.translate.x(10)** in the dialog's **Expression** field.



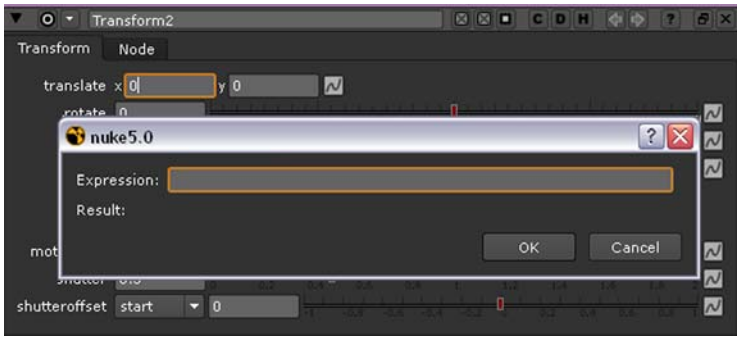
The steps below recap the process for creating a linking expression.

To reference values from another parameter (method #1):

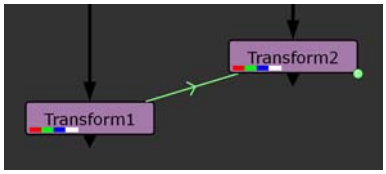
1. Click on the destination parameter (the one which will receive values from another parameter).
2. To display the expression dialog, right-click on the parameter and select **Add expression**



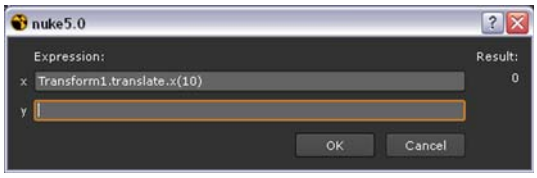
OR type **=** in the parameter field.



3. In the dialog that opens, type the name of the node containing the source parameter and a period. (Each node prominently displays its name on its face.)
4. Follow the name of the node by the source parameter's name and a period. (If you don't know the parameter's name, you can hover over its field in order to see it displayed in a tool tip.)
5. Optionally, type the child parameter's name and a period.
6. Optionally, type a frame number or offset variable in brackets (for example, **(2)** or **(t-2)**) in order to specify the frame or range of frame from which you pull values.
7. Click **OK**. This links the parameters, which turn cyan. In the Node Graph, a green arrow appears between the nodes to indicate that they are linked via an expression.

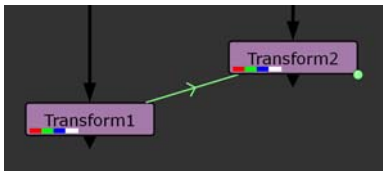


8. To edit the expression later on, right-click on the parameter and select **Edit expression** (or press = on the parameter). You can also click the animation button and select **Edit expression** to edit expressions for all the parameters next to the button.



To reference values from another parameter (method #2):

1. **Ctrl/Cmd**+drag the parameter that has the values you want to use on top of the parameter that will receive these values. This links the parameters, which turn cyan. In the Node Graph, a green arrow appears between the nodes to indicate that they are linked via an expression.



   To view or edit the expression, right-click on the parameter and select **Edit expression**.

2. If you want to link several parameters at the same time, **Ctrl/Cmd**+drag the animation button next to the source parameters on top of the animation button next to the destination parameters.

   To view or edit the expressions used to link the parameters, click the animation button and select **Edit expression**.

To link a parameter driven by an expression:

1. Create the linking expression according to the process described above (method #1).
2. Type the word **expression** in front of the actual expression.
3. Enclose all of the above in square brackets (for example, **[expression Transform1.translate.x]**).

Adding Mathematical Functions to Expressions

You can incorporate mathematical functions into parameters. For example, you might negate an expression in order to invert a tracking curve which you wish to use to stabilise an element (such an expression might resemble the following: **-(Transform1.translate.x)**).

You can also rely on a function to add more complex mathematical operation to your expressions. The table below list all the functions which you may incorporate into Nuke expressions.

Function	Purpose	Operator Usage	Related Functions
abs (x)	Returns the absolute value of the floating-point number x.	x	See also: fabs.
acos (x)	Calculates the arc sine of x; that is the value whose sine is x.	If x is less than -1 or greater 1, asin returns nan (not a number).	See also: cos, cosh, asin, atan.
asin (x)	Calculates the arc sine of x; that is the value whose sine is x.	If x is less than -1 or greater 1, asin returns nan (not a number)>	See also: sin, sinh, acos, atan.
atan (x)	Calculates the arc tangent of x; that is the value whose tangent is x. The return value will be between -PI/2 and PI/2.	x	See also: tan, tanh, acos, asin, atan2.
atan2 (x, y)	Calculates the arc tangent of the two variables x and y. This function is useful to calculate the angle between to vectors.	x, y	See also: sin, cos, tan, asin, acos, atan, hypot.
ceil (x)	Round x up to the nearest integer.	x	See also: floor, trunc, rint.
clamp (x, min, max)	Return x clamped to [min ... max]	x, min, max	See also: min, max.
clamp (x)	Return x clamped to [0.0 ... 1.0]	x	See also: min, max.
cos (x)	Returns the cosine of x	x in radians	See also: acos, sin, tan, cosh.
cosh (x)	Returns the hyperbolic cosine of x, which is defined mathematically as (exp(x) + exp(-x)) / 2.	x	See also: cos, acos, sinh, tanh.

Function	Purpose	Operator Usage	Related Functions
curve (frame)	Returns the y value of the animation curve at the given frame	optional: frame, defaults to current frame	See also: value, y.
degrees (x)	Convert the angle x from radians into degrees	x	See also: radians.
exp (x)	Returns the value of e (the base of natural logarithms) raised to the power of x.	x	See also: log, log10.
exponent (x)	Exponent of x.	x	See also: mantissa, ldexp.
fBm (x, y, z, octaves, lucanarity, gain)	Fractional Brownian Motion. This is the sum of octaves calls to noise(). For each of them the input point is multiplied by pow(lacunarity,i) and the result is multiplied by pow(gain,i). For normal use, lacunarity should be greater than 1 and gain should be less than 1.	x, y, z, octaves, lacunarity, gain	See also: noise, random, turbulence.
fabs (x)	Returns the absolute value of the floating-point number x.	x	See also: abs.
false ()	Always returns 0		See also: true.
floor (x)	Round x down to the nearest integer.	x	See also: ceil, trunc, rint.
fmod (x, y)	Computes the remainder of dividing x by y. The return value is x - n y, where n is the quotient of x / y, rounded towards zero to an integer.	x, y	See also: ceil, floor.

Function	Purpose	Operator Usage	Related Functions
frame ()	Return the current frame number.		See also: x.
from_byte (color component)	Converts an sRGB pixel value to a linear value.	color_component	See also: to_sRGB, to_rec709f, from_rec709f.
from_rec709f (color component)	Converts a rec709 byte value to a linear brightness	color_component	See also: form_sRGB, to_rec709f.
from_sRGB (color component)	Converts an sRGB pixel value to a linear value.	color_component	See also: to_sRGB, to_rec709f, from_rec709f.
hypot (x, y)	Returns the sqrt(x*x + y*y). This is the length of the hypotenuse of a right-angle triangle with sides of length x and y.	x, y	See also: atan2.
int (x)	Round x to the nearest integer not larger in absolute value.	x	See also: ceil, floor, trunc, rint.
ldexp (x)	Returns the result of multiplying the floating-point number x by 2 raised to the power exp.	x, exp	See also: exponent.
lerp (a, b, x)	Returns a point on the line f(x) where f(0)==a and f(1)==b. Matches the lerp function in other shading languages.	a, b, x	See also: step, smoothstep.
log (x)	Returns the natural logarithm of x.	x	See also: log10, exp.
log10 (x)	Returns the base-10 logarithm of x.	x	See also: log, exp.
logb (x)	same as exponent()	x	See also: mantissa, exponent.

Function	Purpose	Operator Usage	Related Functions
mantissa (x)	Returns the normalized fraction. If the argument x is not zero, the normalized fraction is x times a power of two, and is always in the range 1/2 (inclusive) to 1 (exclusive). If x is zero, then the normalized fraction is zero and exponent() Returns zero.	x	See also: exponent
max (x, y, ...)	return the greatest of all values	x, y, (...)	See also: min, clamp.
min (x, y, ...)	return the smallest of all values	x, y, (...)	See also: max, clamp
mix (a, b, x)	same as lerp()	a, b, x	See also: step, smoothstep, lerp
noise (x, y, z)	creates a 3D Perlin noise value. This produces a signed range centred on zero. The absolute maximum range is from -1.0 to 1.0. This produces zero at all integers, so you should rotate the coordinates somewhat (add a fraction of y and z to x, etc.) if you want to use this for random number generation.	x, optional y, optional z	See also: random, fBm, turbulence
pi ()	Returns the value for pi (3.141592654...)		
pow (x, y)	Returns the value of x raised to the power of y.	x, y	See also: log, exp, pow

Function	Purpose	Operator Usage	Related Functions
pow2 (x)	Returns the value of x raised to the power of 2.	x, y	See also: pow
radians (x)	convert the angle x from degrees into radians	x	See also: degrees
random (x, y, z)	creates a pseudo random value between 0 and 1. It will always generate the same value for the same x, y and z. Calling random with no arguments will create a different value on every invocation.	optional x, optional y, optional z	See also: noise, fBm, turbulence
rint (x)	Round x to the nearest integer.	x	See also: ceil, floor, int, trunc
sin (x)	Returns the sine of x	x in radians	See also: asin, cos, tan, sinh
sinh (x)	Returns the hyperbolic sine of x, which is defined mathematically as (exp(x) - exp(-x)) / 2.	x	See also: sin, asin, cosh, tanh
smoothstep (a, b, x)	Returns 0 if x is less than a, returns 1 if x is greater or equal to b, returns a smooth cubic interpolation otherwise. Matches the smoothstep function in other shading languages.	a, b, x	See also: step, lerp
sqrt (x)	Returns the non-negative square root of x.	x	See also: pow, pow2

Function	Purpose	Operator Usage	Related Functions
step (a, x)	Returns 0 if x is less than a, returns 1 otherwise. Matches the step function other shading languages.	a, x	See also: smoothstep, lerp
tan (x)	Returns the tangent of x	x in radians	See also: atan, cos, sin, tanh, atan2
tanh (x)	Returns the hyperbolic tangent of x, which is defined mathematically as sinh(x) / cosh(x).	x	See also: tan, atan, sinh, cosh
to_byte (color component)	Converts a floating point pixel value to an 8-bit value that represents that number in sRGB space.	color_component	See also: form_sRGB, to_rec709f, from_rec709f
to_rec709f (color component)	Converts a floating point pixel value to an 8-bit value that represents that brightness in the rec709 standard when that standard is mapped to the 0-255 range.	color_component	See also: form_sRGB, from_rec709f
to_sRGB (color component)	Converts a floating point pixel value to an 8-bit value that represents that number in sRGB space.	color_component	See also: form_sRGB, to_rec709f, from_rec709f
true ()	Always Returns 1	See also: false	
trunc (x)	Round x to the nearest integer not larger in absolute value.	x	See also: ceil, floor, int, rint

Function	Purpose	Operator Usage	Related Functions
turbulence (x, y, z, octaves, lucanarity, gain)	This is the same as fBm() except the absolute value of the noise() function is used.	x, y, z, octaves, lucanarity, gain	See also: fBm, noise, random
value (frame)	Evaluates the y value for an animation at the given frame.	optional: frame, defaults to current frame	See also: y, curve
x ()	Return the current frame number.		See also: frame
y (frame)	Evaluates the y value for an animation at the given frame	optional: frame, defaults to current frame	See also: value, curve

17 Setting Interface Preferences

Nuke offers a highly customisable interface. This chapter teaches you how you can use the preferences dialog to define the colour, font, and spatial arrangement of all main interface elements.

Displaying the Preferences Dialog

Choose **Edit** > **Preferences**, or press **Shift**+**S**. The preferences dialog appears.

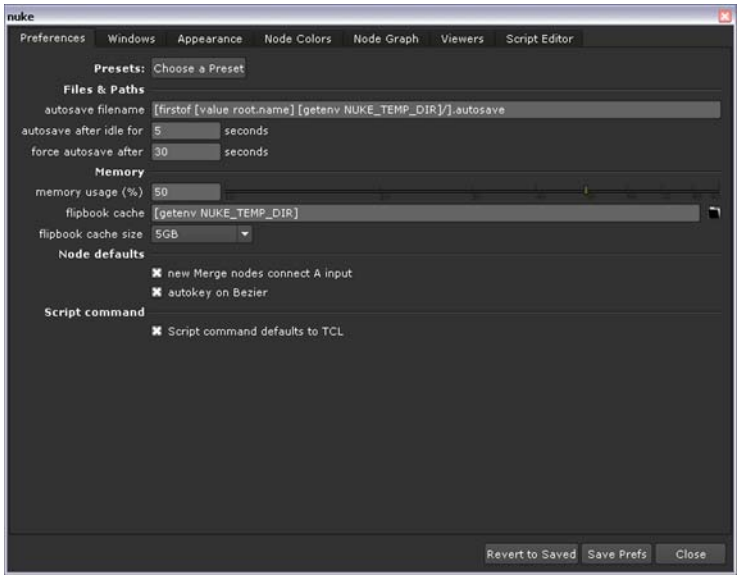


Figure 17-1: Nuke preferences.

Changing Preferences

The function of each preference is described below under *The Available Preference Settings* on page 368.

When you make a change to a preference, in most cases, the interface registers that change immediately (for example, an interface element displays in the new colour).

Saving Preferences

Nuke stores your preference settings in a file called **preferences.nk**, which resides in your Home directory. Each Nuke user can maintain his or her own unique settings.

If you make changes inside the preferences dialog, you need to explicitly save your changes to this file by clicking **Save Prefs**. If you simply close the Preferences dialog, they will only be in effect for your current session. To reset any changes you made and use the preferences saved in preferences.nk, click **Revert to Saved**.

To save your preferences:

Make the desired changes inside the preferences dialog, then click the **Save Prefs** button. Nuke writes the new settings to **preferences5.nk** file, which you can find in the .nuke directory:

- **On Windows**: The .nuke directory can be found under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable - which is generally of the form *drive letter*:\Documents and Settings*login name*\ (Windows XP) or *drive letter*:\Users*login name*\ (Windows Vista).
  To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you will get an error.
- **On Mac OS X**: /Users/*login name*/.nuke
- **On Linux**: /users/*login name*/.nuke

Your new preferences will remain in effect for the current and all subsequent sessions.

Resetting Preferences

To delete your preferences and reset them to default, delete the **preferences5.nk** file. The next time you launch Nuke, it will rebuild the file with the default preferences.

The Available Preference Settings

The preferences dialog is divided into the following tabs:

- **Preferences** - Settings for colour schemes, automatic backup of files, memory usage, certain node defaults, and script command dialog defaults.
- **Windows** - Settings for window positions, tool tips, window snapping, and control panel behaviour.
- **Appearance** - Settings for changing the colours and font on the application interface.
- **Node Colors** - Settings for changing the colours of different nodes and Viewer overlays.
- **Node Graph** - Settings for changing the appearance of the Node Graph (for example, colours, font, background grid usage, arrow size, and Dot node behaviour).
- **Viewers** - Settings for changing the colours, controls, interaction speed, and buffer bit depth of Viewers.
- **Script Editor** - Settings for changing the behaviour and syntax highlighting colours of the Script Editor.

The below tables describe the available preference settings and their functions. Each tab of preferences is described in its own table.

Preferences Tab

Setting	Function
Choose a Preset	Select a predefined colour scheme: **Standard** or **Silver**. You can also create new colour schemes yourself. Do the following: 1. Go to the **Appearance** tab and adjust the colours until you are happy with them. Click **Save Prefs**. 2. Make a copy of the preferences5.nk file and rename it to UI_*name*.tcl (where *name* is the name of your colour scheme). You can find preferences5.nk in the following location: • On Windows: In the .nuke directory, which can be found under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable – which is generally of the form "*drive letter*:\Documents and Settings*login name*\" (Windows XP) or "*drive letter*:\Users*login name*\" (Windows Vista). To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error. • On Mac OS X: /Users/*login name*/.nuke • On LInux: /users/*login name*/.nuke 3. Put this file into the plug-in path. For more information on the plug-in path, see *Environment Variables* on page 406. When you next launch Nuke, your new colour scheme is listed as one of the predefined colour schemes.
Files & Paths	
autosave filename	Define where and under what name Nuke saves your automatic backup files. By default, the files are saved with the extension *.autosave* in the same folder as your project files. To change this, enter a full directory pathname in the autosave filename field. You can use *[value root.name]* to refer to the full script pathname, and *[file tail [value root name]]* to just refer to the filename with its extension.
autosave after idle for	Define how long (in seconds) Nuke waits before performing an automatic backup after you have left the system idle (that is, haven't used the mouse or the keyboard). If you set the value to 0, automatic backups are disabled.
force autosave after	Define how long (in seconds) Nuke waits before performing an automatic backup regardless of whether the system is idle. If you set the value to 0, forced automatic backups are disabled.

Setting	Function
Memory	
memory usage (%)	Set the amount of RAM that Nuke uses for processing images. Generally, the default setting of 50% gives a good trade off between performance and interactivity.
disk cache	Nuke to saves all recent images displayed in the Viewer for fast playback. Using this control, you can specify where you want Nuke to save these images. Pick a local disk (for example, c:/temp), preferably with the fastest access time available. It's also important to leave enough space for the maximum disk cache size (defined below). The environment variable NUKE_DISK_CACHE can be used to override this setting.
disk cache size	Select the maximum size the disk cache can reach. Ensure there is enough space on the disk for this to be reached. The environment variable NUKE_DISK_CACHE_GB can be used to override this setting.
Node defaults	
new Merge nodes connect A input	When this is checked, any Merge nodes you add are connected to the currently selected node via the A input of the Merge node. When this is NOT checked, the B input of the Merge node is used instead.
autokey on Bezier	When this is checked, the **autokey** control in the Bezier node properties is enabled by default, and Bezier shapes are automatically saved as key shapes. When this is NOT checked, the **autokey** control is disabled by default, and Bezier shapes are not automatically saved as key shapes.
Script command	
Script command dialog defaults to TCL	Select which scripting language the script command dialog defaults to when you select **File** > **Script Command**. When this is checked, the dialog defaults to TCL. When this is NOT checked, it defaults to Python.

Windows Tab

Setting	Function
window Positions: Clear	When you rearrange floating windows, such as the colour picker window, Nuke remembers their position the next time you open them. You can use this control to clear the remembered positions.
tooltip delay	Define how long (in seconds) you need to hover the cursor over a control before Nuke displays its tool tip. If you set this value to 0, no tool tips are displayed.

Setting	Function
floating window hint	This preference only has an effect on Linux and can be used to fix problems with floating windows. What you should set it to depends on the window manager you are using (GNOME or KDE): • **Utility (GNOME)** - Choose this if you are using the GNOME window manager. If you haven't specifically selected to use KDE, you're most likely using GNOME. • **Normal (KDE)** - Choose this if you are using the KDE window manager. If this option is not used on KDE, all the floating windows vanish when you move focus away from Nuke.
Script Loading	
re-open viewers when loading saved scripts	When this is checked and you open a saved script, any Viewers in the script are opened in the Viewer pane. When this is NOT checked, you need to open the Viewers manually by double-clicking on them in the Node Graph.
use window layout from saved scripts	When this is checked, Nuke opens saved scripts with the window layout they were saved with. When this is NOT checked, Nuke opens saved scripts with the default layout.
Window Snapping	
snap when moving windows	When this is checked and you move floating windows, the windows snap to screen edges and other floating windows, making it easy to place them right next to each other. When this is NOT checked and you move floating windows, the windows do not snap to anything. On Linux, window snapping may not work. However, most Linux window managers let you do window snapping if you hold down **Shift** while moving the window.
snap if parallel without touching	When this is checked, all floating windows' edges are considered to extend infinitely outward, so that you can easily align the windows even if they aren't close to each other. For example, say you have one floating window on the left and another on the right. When you move either of these up or down, their top and bottom edges snap to align with the top or bottom edge of the other window. When this is NOT checked, window snapping is restricted to nearby windows.
threshold	Define how close to each other (in pixels) the windows have to be for them to snap together.

Setting	Function
Control Panels	
new panels go to	Select where you want control panels to appear when you add a new node or double-click on an existing node in the Node Graph: • **own window** - Each new control panel appears in its own floating window. • **top of control panel bin** - Each new control panel appears on top of the Properties Bin. • **bottom of control panel bin** - Each new control panel appears in the bottom of the Properties Bin.
max panels in bin	Define the maximum number of control panels that can appear in the Properties Bin at the same time. When you're working, you can override this setting using the field on top of the Properties Bin. However, the number saved in the preferences is used as the default whenever you launch Nuke.
reopen acts like new panel	When this is checked and you reopen a floating control panel, Nuke opens the panel in the same place as a new panel, even if you moved the panel to a new location earlier. When this is NOT checked and you reopen a floating control panel, Nuke remembers where the panel was located when it was last closed and opens it in that position.
double-click moves panel	When this is checked, you can double-click on a node whose control panel is already open to move the control panel to the place where the new panels go. When this is NOT checked, double-clicking on a node whose control panel is already open does change the panel's position but selects the panel and (in the case of floating panels) moves in on top of other panels.
close Control Panel Bin when empty	When this is checked, the Properties Bin is closed when the last control panel in the Bin is closed. When this is NOT checked, the Properties Bin remains open when the last control panel in the Bin is closed.
expand/collapse panels in bin to match selection	When this is checked, only the control panels of the nodes that are selected in the Node Graph are opened. All unselected nodes will have their control panels collapsed. This only applies to the control panels in the Properties Bin. When this is NOT checked, control panels are not opened or closed based on the selection in the Node Graph.
input button does	Select what happens when you click the input button on top of a control panel and select one of the node's inputs: • **select input node only** - an input of the node is selected in the Node graph. • **scroll node into view** - an input of the node is selected in the Node Graph and, if the input is outside the currently displayed area of the Node Graph, the displayed area is adjusted to include the input node. • **center node** - an input of the node is selected and positioned in the centre of the Node Graph.

Setting	Function
show dialogs under the cursor	When this is checked, pop-up dialogs appear in the current position of the cursor. When this is NOT checked, pop-up dialogs appear in the middle of the Nuke application window.
show menu with previous item under the cursor	When this is checked, right-click menus are opened with the previously selected item under the cursor. When this is NOT checked, right-click menus are opened with nothing under the cursor.

Appearance Tab

Setting	Function
Colors	
Background	Change the background colour of most user interface elements (menus, tool bars, panes, control panels, Viewers, and pop-up dialogs). To set the colour back to default (dark grey), right-click on the button and select **Set color to default**.
Base	Change the colour of input fields, the input pane of the Script Editor, and the left part of the Curve Editor. To set the colour back to default (light grey), right-click on the button and select **Set color to default**.
Highlight	Change the colour of the highlighting that appears when you hover the cursor over a control, select a file or folder in the File Browser, or scrub to a new frame on the timeline. To set the colour back to default (light orange), right-click on the button and select **Set color to default**.
Label	Change the colour of labels and text on the application interface. Note that this does not set the colour of the labels on the nodes in the Node Graph. To set the colour back to default (white), right-click on the button and select **Set color to default**.
Button	Change the colour of buttons and pulldown menus. To set the colour back to default (medium grey), right-click on the button and select **Set color to default**.
Animated	Change the colour that indicates a control has been animated. To set the colour back to default (cyan), right-click on the button and select **Set color to default**.
Keyframe	Change the colour that indicates a key frame has been set. To set the colour back to default (bright cyan), right-click on the button and select **Set color to default**.
Fonts	
label font	Change the type, weight, angle, and size of the font used on the application interface.

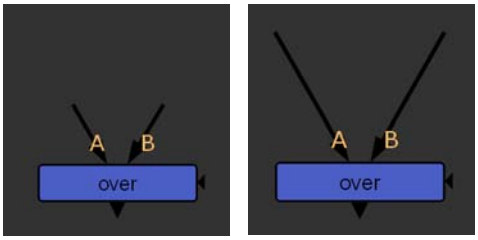
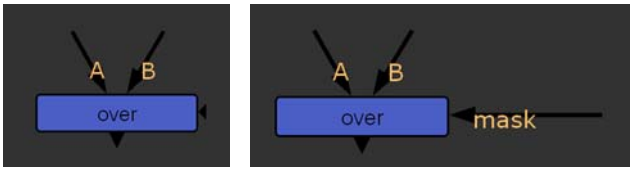
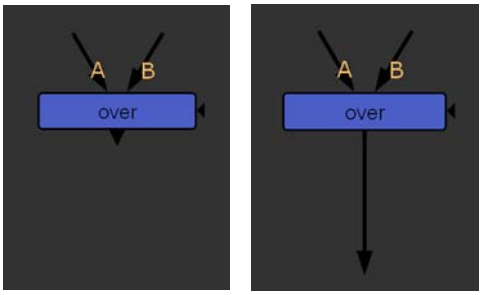
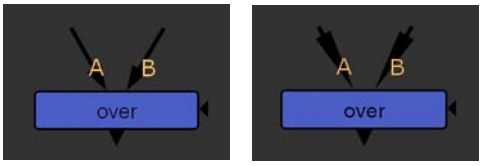
Node Colors Tab

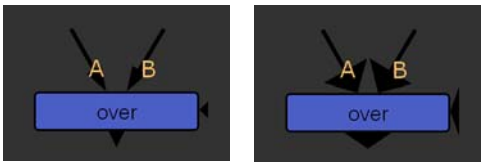
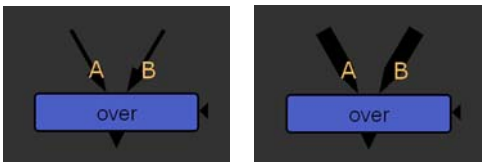
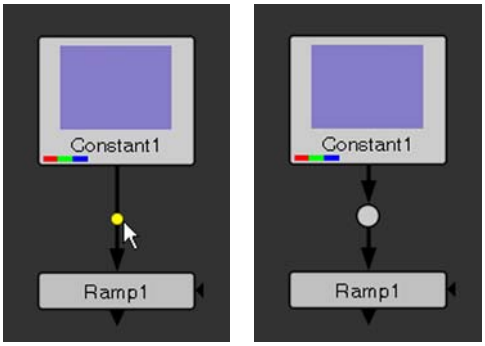
Setting	Function
autocolor	Check this to colour nodes based on the class they belong to. If you don't check this, the All Others colour is used on all nodes.
Shade Nodes	Check this to have the nodes in the Node Graph shaded. If you don't check this, no shading is used.
Beziers, Drawing Ops, Colour Ops, Time Ops, Channel ops, Merge ops, Keyer ops, Write ops, 3D ops, Filters, 2D Xform	Change the colour of the nodes that belong to each group. To set the colour back to default, right-click on the colour button on the right and select **Set color to default**. You can also change which nodes belong to each group by entering the names of the nodes in the text input fields.
User 1, User 2	You can use these controls to create a group of gizmos and change their colour in the Node Graph. List the names of the gizmos (without the .gizmo extension) in the input field, and use the colour button to change their colour in the Node Graph. To set the colour back to default (light grey), right-click on the colour button on the right and select **Set color to default**.
All Others	Change the colour of any nodes that do not belong to the node groups above. To set the colour back to default (light grey), right-click on the button and select **Set color to default**.
Text	Change the colour of the labels and text in the Node Graph (for example, the labels on nodes and the notes you create using StickyNote nodes). To set the colour back to default (black), right-click on the button and select **Set color to default**.
Selected	Change the colour that is used to indicate that nodes are selected. To set the colour back to default (yellow), right-click on the button and select **Set color to default**.
GL Color	Change the colour of the Viewer overlays. To set the colour back to default, right-click on the button and select **Set color to default**.

Node Graph Tab

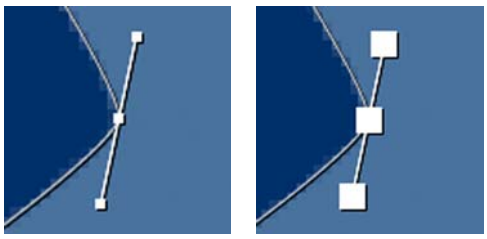
Setting	Function
autolabel	Check this to automatically label nodes with channel/mask information. If you don't check this, nodes will only show the file name or the node name.
highlight running operators	Check this to highlight any nodes whose output is currently being calculated.
node name background	When a node is selected and the node's name is too long to fit inside the node, a background is drawn behind the name to improve legibility. You can use this control to set the intensity of the background, from 0 (no background) to 1 (fully opaque background).

Setting	Function
label font	Change the type, weight, angle, and size of the font used on labels in the Node Graph.
tile size (WxH)	Change the width and height (in pixels) of the nodes in the Node Graph.
snap to node	When this is checked and you move a node in the Node Graph, the node snaps to a position that aligns it horizontally and vertically with its input and output nodes.
grid size (WxH)	Define the width and height (in pixels) of the cells in the background grid that you can display on the Node Graph. To see the grid, check **show grid** below.
snap to grid	When this is checked and you move a node in the Node Graph, the node snaps to a position that lines it up with the background grid lines. To see the grid, check **show grid** below.
show grid	Show a background grid on the Node Graph.
snap threshold	Define the maximum number of pixels to jump by when snapping nodes to other nodes or grid lines.
Colors	
Node Graph	Change the background colour of the Node Graph. To set the colour back to default (dark grey), right-click on the button and select **Set color to default**.
Overlay	Change the colour of the grid that you can have appear on the Node Graph if you check **show grid** below. To set the colour back to default (light grey), right-click on the button and select **Set color to default**.
Elbow	Change the colour of the "elbows" that appear on arrows when you press **Ctrl/Cmd** on the Node Graph, and that you can click to insert Dot nodes. To set the colour back to default (yellow), right-click on the button and select **Set color to default**.
Arrows	
Left arrow button	Change the colour of arrows pointing left in the Node Graph. To set the colour back to default (black), right-click on the button and select **Set color to default**.
Right arrow button	Change the colour of arrows pointing right in the Node Graph. To set the colour back to default (black), right-click on the button and select **Set color to default**.
Up arrow button	Change the colour of arrows pointing up in the Node Graph. To set the colour back to default (black), right-click on the button and select **Set color to default**.
Down arrow button	Change the colour of arrows pointing down in the Node Graph. To set the colour back to default (black), right-click on the button and select **Set color to default**.
expression arrows	Change the colour of the arrows that indicate nodes are connected via an expression. To set the colour back to default (green), right-click on the button and select **Set color to default**.
enable	Check this if you want to display expression arrows in the Node Graph.
clone arrows	Change the colour of the arrows that indicate nodes have been cloned. To set the colour back to default (orange), right-click on the button and select **Set color to default**.

Setting	Function
enable	Check this if you want to display clone arrows in the Node Graph.
unconnected top input arrow length	Adjust the length of the unconnected input arrows that appear on top of nodes in the Node Graph. By default, this value is set to 35 pixels. The maximum value is 70. 
unconnected left/right input arrow length	Adjust the length of the unconnected arrows that appear on the sides of some nodes in the Node Graph. This affects any mask inputs and extra Viewer or Scene node inputs, for example. By default, this value is set to 4 pixels. The maximum value is 70. 
unconnected output arrow length	Adjust the length of unconnected output arrows in the Node Graph. By default, this value is set to 8 pixels. The maximum value is 70. 
arrow head length	Adjust the length of arrow heads in the Node Graph. By default, this value is set to 12 pixels. There is no maximum value. 

Setting	Function
arrow head width	Adjust the width of arrow heads in the Node Graph. By default, this value is set to 8 pixels. There is no maximum value. 
arrow width	Adjust the width of the arrows in the Node Graph. By default, this value is set to 2 pixels. There is no maximum value. 
allow picking of connected arrow heads	When this is checked, you can click on an arrow head and drag it to a new location. When this is NOT checked, connected arrow heads are locked into place, and you can only change the connections by moving the arrow tails.
allow picking of arrow elbows to create Dots	When this is checked, you can press **Ctrl** (**Cmd** on a Mac) on the Node Graph to display yellow "elbows" on the Node Graph arrows and click on these to insert Dot nodes.  If you **Ctrl**/**Cmd**+**Shift**+**click** on an elbow, the new Dot node is branched off to a new arrow rather than inserted in the existing arrow. When this setting is NOT checked, adding Dot nodes in this manner is not possible.
drag-to-insert only works near middle of arrows	When this is NOT checked, you can insert nodes in between other nodes by dragging them over any point of the connecting arrow. When this is checked, you can only insert nodes in between other nodes in the above manner by dragging them over the middle point of the connecting arrow.
size of dots	Adjust the size of Dot nodes. By default, the value is set to 1.

Viewers Tab

Setting	Function
2D bg	Change the background colour of the 2D Viewer. To set the colour back to default (black), right-click on the button and select **Set color to default**.
2D fg	Change the colour of borders and text in the 2D Viewer. To set the colour back to default (light grey), right-click on the button and select **Set color to default**.
3D bg	Change the background colour of the 3D Viewer. To set the colour back to default (black), right-click on the button and select **Set color to default**.
3D fg	Change the colour of borders and text in the 3D Viewer. To set the colour back to default (light grey), right-click on the button and select **Set color to default**.
handle size	Adjust the size of the square control handles that appear on the Viewer for some operations, such as transformations, warps, and Bezier draws. By default, this value is set to 5. 
icon size	Adjust the size of the 2D transformation overlay, 3D camera, 3D object normals, and 3D axis on the Viewer. By default, this value is set to 50.
icon scaling	Adjust how much the scale of display affects the size of the 2D transformation overlay, 3D camera, and 3D axis. When this is set to 0, these controls are always drawn the same size, regardless of the zoom level. When the value is set to 1, the controls scale with the displayed image or 3D scene when you zoom in or out. Intermediate values mix this so that the controls do scale, but not as much as the image does. This gives an optical illusion that you are zooming in or out without making the controls unusably small or large.
3D control type	Select the navigation control scheme you want to use in the 3D Viewer: Nuke, Maya, Houdini, or Lightwave.
texture size	Select the size of the texture maps for the OpenGL preview of 3D objects and 2D transformations. The default size is 512x512.
object interaction speed	Set how fast mouse movements rotate and translate 3D axis and cameras. The lower the value, the finer the movements. The default value is 0.1.
camera interaction speed	Set how fast mouse movements tumble and roll the 3D view in the Viewer. The lower the value, the finer the movements. The default value is 1.

Setting	Function
delete Viewer node when Viewer window is closed	When this is checked, Viewer nodes are deleted from the Node Graph when you close the associated Viewers in the Viewer pane. When this is NOT checked, closing Viewers in the Viewer pane does not affect Viewer nodes in the Node Graph.
Viewer buffer bit depth	Set the default Viewer OpenGL buffer depth: • **float** - Uses a full float texture and applies the LUT, gain, and gamma in the GPU. • **half-float** - Converts to 16-bit (half) float and applies the LUT, gain, and gamma in the GPU. • **byte** - Uses 8-bit with error diffusion and applies the LUT, gain, and gamma in software. This is the default. **Note**: When Nuke is set to use the 8-bit (software) path, the gain, gamma, and LUTs are applied in the CPU, and we also do error diffusion in the conversion from the internal floating point image data to 8-bit. This is to avoid the appearance of banding due to the reduced bit depth. Without this, artists often mistakenly assume the banding they see is actually present in their image data and try to blur to remove it, softening the image unnecessarily. However, because Nuke works internally at floating point, there is little likelihood of banding actually being present. In the half-float and float (hardware) paths, we apply the gain, gamma, and LUTs in the GPU, but we are currently unable to apply decent error diffusion here. This means that when the graphics card converts to the lower bit depth at output to the monitor, the appearance of banding can occur. When banding is visible in the image, artists are advised to switch the Viewer to 8-bit (software) mode and/or sample pixel values. This way, they can confirm that the image is not actually banding, and that it's merely an artifact of the limited bit depth of current display hardware before attempting to mitigate the banding.

Script Editor Tab

Setting	Function
Input	
clear input window on successful script execution	When this is checked, any successfully executed Python statements disappear from the input pane of the Script Editor and appear in the output pane. When this is NOT checked, all statements stay in the input pane of the Script Editor, even if they were successfully executed.
echo python commands to output window	Check this to have all Python commands executed by yourself or Nuke appear in the output pane of the Script Editor. This way, you can, for example, select a node from the Toolbar and have the corresponding Python command displayed in the output pane.

Setting	Function
Syntax Highlighting	
keywords	Change the colour that's used to highlight any words that are Python's keywords (for example, print and import) in the Script Editor. To set the colour back to default (green), right-click on the button and select **Set color to default**.
string literals (double quotes)	Change the colour that's used to highlight strings (inside double quotation marks) in the Script Editor. To set the colour back to default (red), right-click on the button and select **Set color to default.**
string literals (single quotes)	Change the colour that's used to highlight strings (inside single quotation marks) in the Script Editor. To set the colour back to default (cyan), right-click on the button and select **Set color to default.**
comments	Change the colour that's used to highlight comments (anything beginning with #) in the Script Editor. To set the colour back to default (yellow), right-click on the button and select **Set color to default.**
Font	Change the type, weight, angle and size of the font used in the Script Editor.

18 The Script Editor and Python

Have you ever thought of something you'd absolutely love to be able to do with Nuke but that does not seem to have been a priority for Nuke developers? Or would you just die to be able to automate some of the more tedious procedures, so that they'd take care of themselves while you aren't even at your desk? Well, we have just the answer for you: Python, one of the two scripting languages Nuke supports (the other is TCL).

This chapter describes how you can use the Script Editor for executing Python scripts, gives you examples of those scripts, teaches you how to automate your scripts, and directs you to sources of more information.

Workflow

The workflow for using Python in Nuke is generally the following:

1. Enter Python statements in Nuke's Script Editor to perform the actions you want to perform.
2. Save your script with the extension .py in a directory that is contained in the sys.path variable.
3. Later, when you want to execute the same statement sequence, import the .py file into Nuke's Script Editor. Nuke executes the statements in the specified order.

Note

You can also run an interactive Python session outside of Nuke. However, on Mac OS X the Python interpreter and other applications that are bundled with Nuke (such as IDLE) can only be run from inside the directory where they are located.

Using the Script Editor

You type Python scripts into Nuke's Script Editor.

To open the Script Editor:

To open the Script Editor, click on one of the content menus and select **Script Editor** from the menu that opens.

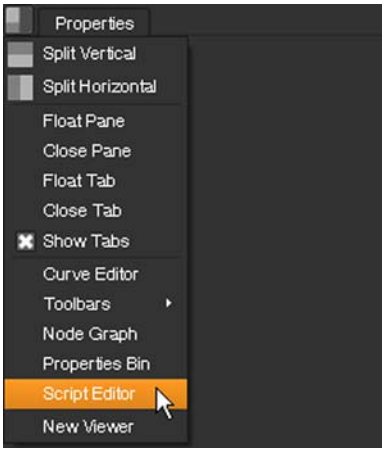


Figure 18-1: Opening the Script Editor.

Input and Output Panes

The Script Editor is divided into two parts. You use the lower part (input pane) to type in and execute your Python statement, and when you have done so, statements and their outputs appear in the upper part of the editor (output pane). Successfully executed statements are followed by a hash mark (**#**).

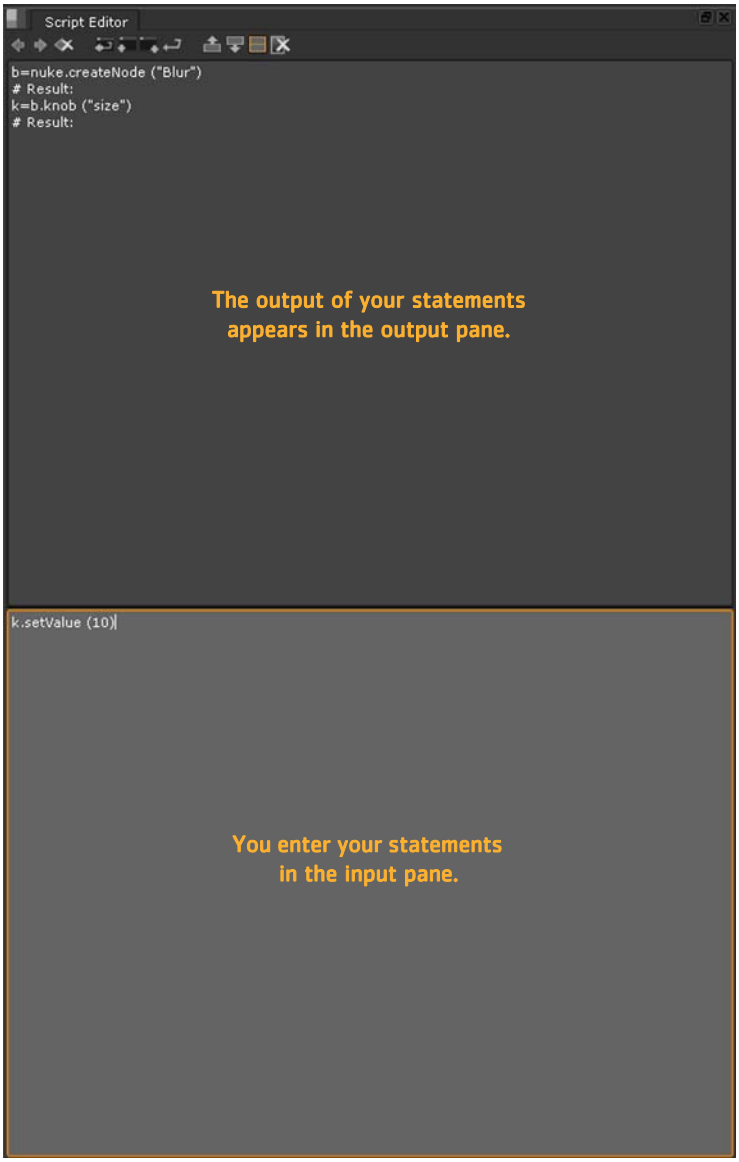


Figure 18-2: The two parts of the Script Editor.

To hide the output or input pane, click the **Show input only** or **Show output only** button on top of the Script Editor.

To show both panes again, click the **Show both input and output** button.

To enter a statement:

1. Click on the input pane of the editor to insert the cursor there.
2. Type in your statement. To use the usual editing functions, such as copy and paste, right-click on the editor and select the desired function.

When entering the statement, you'll notice that any words that are Python's keywords (such as *print* and *import*) turn green, while strings (basically, anything in quotation marks) become either red or cyan. Comments are shown in yellow.

If you like, you can change these colours and the font on the **Script Editor** tab of the Preferences dialog. To open the preferences, press **Shift**+**S**.

3. If your statement includes several lines or you want to enter several statements at once, press **Return** to move to the next line.
4. To execute the statement, click the **Run the current script** button on the top of the Editor, or press **Ctrl**+**Return** (**Cmd**+**Return** on a Mac).

Tip

You can also execute statements by pressing **Ctrl**+**Enter** (**Cmd**+**Enter** on a Mac) on the numeric key-pad.

By default, successful statements disappear from the input pane, and appear in the output pane. However, if you want all statements to stay in the input pane after they are executed, you can do the following:

1. Press **Shift**+**S** to open the Preferences dialog.
2. Go to the **Script Editor** tab.
3. Uncheck **clear input window on successful script execution**.
4. Click **Close** to save the preference for the current project only, or **Save Prefs** to save the preference for the current and future projects.

If you enter an invalid statement, Nuke produces an error in the output pane of the Script Editor, leaving the invalid statement in the input pane. Correct the statement and execute it again until you get it right.

Note

Sometimes you may get an error if you copy and paste statements into the Script Editor from another source, like an e-mail. This may be caused by the mark-up or encoding of the source you copied the statement from. To fix the problem, re-enter the statement manually.

If you want to have all executed Python commands appear in the output pane of the Script Editor, open the Preferences dialog (press **Shift**+**S**), go to the **Script Editor** tab, and check **echo all commands to output window**. This applies to both commands executed by yourself and by Nuke. For example, if you select a node from the Toolbar, the corresponding Python command is displayed in the output pane. This does not apply to all actions you take in the

graphical user interface, however, but only those that are performed by executing Python script commands.

To only execute part of a script, enter the script in the input pane and select the part you want to execute. Press **Ctrl+Return** (**Cmd+Return** on a Mac). Nuke runs the selected part of the script, leaving the script in the input pane.

To repeat a statement, click the **Previous Script** button on top of the Editor to move back to the previous statement. You can do this several times until you reach the statement you want to repeat. To execute the statement again, press **Ctrl+Enter** (**Cmd+Enter** on a Mac).

To decrease or increase the indentation of text in the input window, select the text and press **Ctrl/Cmd+Shift+[** or **Ctrl/Cmd+Shift+]**.

To move through or clear the script history:

In addition to stepping backwards through the history of your script, you can also step forwards. Click the **Next script** button to move forward through your statements.

To clear the history, click the **Clear history** button.

To clear the output pane:

Click the **Clear output window** button (or press **Ctrl/Cmd+Backspace**).

So, now you know how to use the editor. But what do you do with this knowledge if you don't know any statements? Not much. Next, we look at some example scripts you can enter in the editor.

Example Scripts

As you will soon notice after trying to enter a few scripts yourself, the scripts are case-sensitive and will only work if you enter them exactly right. There's a bit of flexibility when it comes to quotation marks and spaces, though. You can use both single and double quotes in your statements, and you don't necessarily need to add a space before the brackets like we have done in the following examples.

Creating Nodes and Setting Their Parameters

This section explains how to create nodes and set their parameters using Python.

To create a node, enter:

nuke.nodes.*nodename* (...)

where *nodename* is the name of the node you want to create (for example, Bezier or Blur). You can only use this statement if you know the name of the node you want to create and can enter it explicitly in the statement.

For example, to add a Blur node, enter:

nuke.nodes.Blur ()

There is also another statement you can use to create nodes in Nuke. That is the **nuke.createNode (...)** statement. It can be useful if the name of the node is only available to you as a variable or is returned by a function or method.

If you wanted to create a Blur node using this statement, you would enter:
nuke.createNode ("Blur")

Note

You may encounter problems if you attempt to use the **nuke.nodes.*nodename* (...)** or **nuke.createNode (...)** statements to call a file that does exist in your [Nuke directory]/plugins folder but is NOT normally used to create a node or called directly. For example, entering **nuke.nodes.drop ()** or **nuke.createNode("drop")** causes Nuke to try and call the drop.tcl file, but because no node called Drop exists, this may result in a crash.

In general, any files in the plugins directory whose name does not start with a capital letter do not create nodes and should not be used with the **nuke.nodes.*nodename*** or **nuke.createNode** statements.

When using the **nuke.nodes.*nodename* (...)** statement to create nodes, you can also add optional named arguments inside the parenthesis, separated by commas. You can use these arguments to set parameter values for the node, rename the node, set the nodes inputs, and (in the case of Read nodes) specify where to load an image from. Setting parameter values, renaming nodes, and creating Read nodes are described below, whereas setting a node's inputs is discussed later in this chapter, under *Connecting Nodes and Setting Their Inputs* on page 388.

To create a node and set a parameter value, enter:

nuke.nodes.*nodename* (*parameter*=*value*)
where

- ***nodename*** is the name of the node you want to add
- ***parameter*** is the name of the control whose value you want to set
- ***value*** is the value you want to give to the control.

For example, if you wanted to create a Blur node and set its size parameter to 10, you would need to enter **nuke.nodes.Blur (size=10)**.

You can check the results in the Blur node's properties panel, where the value in the **size** field should be set to 10.

To create a node and rename it, enter:

nuke.nodes.*nodename* (name="*newname*")
where

- ***nodename*** is the name of the node you want to add
- ***newname*** is the name that you want to give to the node.

For example, if you wanted to create a Camera node but call it *Projection_Cam*, you would enter **nuke.nodes.Camera (name="Projection_Cam")**.

To create a Read node, enter:

nuke.nodes.Read (file="*filepath\filename.ext*")
where ***filepath\filename.ext*** represents the path to the image you want to read in. For example, if you wanted to read in **myimage.cin** from **C:\Temp**, you would enter **nuke.nodes.Read (file="C:\Temp\myimage.cin")**.

Assigning Variables

Since you are likely to want to play with the node after creating it, it makes sense to assign the node to a variable. Then, in your subsequent statements, you can use the variable to refer to the node.

To add a node and assign it to a variable, enter:

***variable*=nuke.nodes.*nodename* (...)**
where ***variable*** represents any character or string of characters you want to use as the variable, and ***nodename*** the name of the node you want to create.

For example, you could enter:
b=nuke.nodes.Blur ()
Here, you are adding a Blur node and deciding to call it **b** in your subsequent statements. Of course, you could call it anything you like. If you were adding several Blur nodes to your script, you might want to call them **b1**, **b2**, and **b3**, for example.

Now, say you created a Blur node and then decided to set its **size** parameter to 10. If you have assigned the node to a variable (for example, **b**), you can do this by entering **b["size"].setValue (10)**.

Adding Parameters to Node Controls

We've now looked at setting parameter values for the existing parameters of a node. But what should you do if you want to add a completely new parameter to a node? You need to use, for example, the following statements:
b = nuke.nodes.*nodename* (...)
k = nuke.Array_Knob("*name*", "*label*")
b.addKnob(k)
where

- ***nodename*** represents the name of the node you want to add the new parameter to.
- ***name*** represents a unique name you give the parameter to reference it using scripts and expressions.
- ***label*** represents the label shown next to the control in the node's properties panel.

Let's say you enter the following lines:
b = nuke.nodes.Blur ()
k = nuke.Array_Knob("myctrl", "My Control")
b.addKnob(k)

Using these statements, you add a Blur node and assign it to the variable b. You then create an input field parameter, decide to call it "myctrl" in references and "My Control" on the user

interface, and assign the parameter to the variable k. Finally, you add the parameter (k) to the Blur node's (b) controls.

If you wanted to create a slider control rather than an input field, you'd need to use the following statements (replacing **Array_Knob** with **WH_Knob**):

```
b = nuke.nodes.Blur ()
k = nuke.WH_Knob("myctrl", "My Control")
b.addKnob(k)
```

The following lines, instead, would produce a checkbox control:

```
b = nuke.nodes.Blur ()
k = nuke.Boolean_Knob("myctrl", "My Control")
b.addKnob(k)
```

Now you have a control, but it has no tool tip. Let's add one. Enter the following statement:

```
k.setTooltip('My tooltip')
```

This adds a tool tip with the words *My tooltip* to the control created earlier and assigned to variable k.

Note

To see the tool tip, you may need to close and reopen the node's properties panel.

Connecting Nodes and Setting Their Inputs

By default, the nodes you add to your node tree are connected to the last selected node. However, you can also use a script to set an input for a node.

Let's imagine you want to add two Read nodes and a Merge node (in this case, an over node) to your node tree, and connect the Read nodes to the over node's **A** and **B** inputs (numbers 1 and 0), like this:

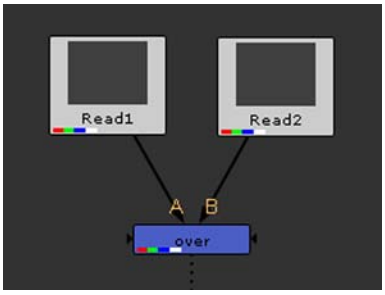


Figure 18-3: Read nodes and a Merge (over) node.

You need to add the Read nodes and the over node and specify the over node's inputs using, for example, the following statements:

- **r1=nuke.nodes.Read (file="*filepath\filename.ext*")**
- **r2=nuke.nodes.Read (file="*filepath\filename.ext*")**
- **m=nuke.nodes.Merge (inputs=[r2, r1])**

where ***filepath\filename.ext*** represents the locations and names of the images you want to read in. The last statement creates a Merge node and sets r2 (the second Read node) and r1 (the first Read node) as its inputs.

Setting Default Values for Controls

You can set default values for controls in any nodes that belong to the same class. After a default value is set, all controls with matching names will default to this value. To set a default value, use the following statement:
nuke.knobDefault()

For example, if you want the **size** control on Blur nodes to default to 20, you can use the following statement:
nuke.knobDefault("Blur.size", "20")

If you then wanted the last frame value of the **frame range** controls in the Project Settings to default to frame 200, you could use the following statement:
nuke.knobDefault("Root.last_frame", "200")

Notice that you need to start the node class with a capital letter (for example, **Root** rather than **root**).

It is also important that you add these statements in your init.py file rather than menu.py. This ensures that they are set for command line start-up as well as the graphical user interface (GUI). If these statements are not processed until menu.py (GUI start-up), then nodes created from the command line Nuke Python prompt (that is, starting as **nuke -t**) will not have the defaults applied.

The init.py file is a file that is run before menu.py usually to set things that are GUI independent. It should be located in your plug-in path directory (for more information on plug-in path directories, see *Environment Variables* on page 406). If the init.py file does not yet exist, simply create a text file and name it init.py.

Rendering with the Write Node

Let's imagine you've added a Write1 node into your script. Now, you want to render every other frame of the frames from 1 to 35.

To render a single Write node, enter:

nuke.execute (*"name",start,end,incr*)
where

- ***name*** represents the name of the write node, for example **Write1**
- ***start*** represents the first frame you want to render
- ***end*** represents the last frame you want to render
- ***incr*** represents the increment you want to use when rendering. For example, if you used 3 here, Nuke would render every third frame.

In our example, you would enter
nuke.execute ("Write1",1,35,2)
or
nuke.execute ("Write1",start=1,end=35,incr=2)

Tip

Instead of **nuke.execute (name,start,end,incr)**, you can also use **nuke.render (name,start,end,incr)**. Both statements perform the same action.

To render multiple Write nodes or ranges, enter:

nuke.executeMultiple ((*variable*,), ([*start,end,incr*],))
where

- ***variable*** represents the variables of the write nodes
- ***start*** represents the first frames you want to render
- ***end*** represents the last frames you want to render
- ***incr*** represents the increments you want to use when rendering. For example, if you used 3 here, Nuke would render every third frame.

Listing a Node's Controls

Let's try something a little more complicated and list all the controls in a node's properties panel.

To list a node's controls, enter:

```
for i in range (getNumKnobs()):
 print knob (i).name()
```

Make sure you enter this compound statement on two separate lines and indent the second line.

For example, to list the controls of the Blur node you added earlier and assigned to variable **b**, you would enter:

```
for i in range (b.getNumKnobs()):
 print b.knob (i).name()
```

As a result of this, Nuke lists the Blur node's controls, displaying them in the output pane of the Script Editor.

Undoing and Redoing Actions

Oops. Made a mistake? Let's undo it.

To undo actions, enter:

```
nuke.undo()
```

If you undo an action by accident, redoing it is just as easy.

To redo actions, enter:

```
nuke.redo()
```

Frame Navigation

To use the frame navigation buttons of the currently active Viewer, enter the following statement:

```
nuke.activeViewer().frameControl(i)
```

where **i** is an integer that indicates the frame navigation button you want to execute. You can replace **i** with the following:

- **-6** to go to the first frame.
- **-5** to go to the previous key frame.
- **-4** to step back by increment.
- **-3** to go back to the previous key frame or increment, whichever is closer.
- **-2** to step back one frame.
- **-1** to play the sequence backward.
- **0** to stop playback.
- **+1** to play the sequence forward.
- **+2** to step forward one frame.
- **+3** to go to the next key frame or increment, whichever is closer.
- **+4** to step forward by increment.
- **+5** to go to the next key frame.
- **+6** to go to the last frame.

You can also assign these buttons to a hot key. For example, to assign the play button to the Up arrow key, do the following:

1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Add the following entry in your menu.py:

```
def play():
  v = nuke.activeViewer()
  if v:
    v.play( 1 )


menubar = nuke.menu("Nuke");
m = menubar.addMenu("&File")
m.addCommand("@;Play", "play()", "Up")
```

This assigns the hot key to an invisible menu item (@ before the name of the menu item makes the item invisible).

Overriding the Creation of a Particular Node

Sometimes, you may want to override the creation of a particular node. For example, Nuke includes two versions of the Merge node: Merge and Merge2. By default, Merge2 is inserted when you select **Merge** > **Merge**. However, if you'd prefer to use Merge, you can override the creation of the Merge node so that Nuke creates the Merge node by default. To do so, follow these steps:

1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Add the following entry:

```
class MyCustomNodes():
  def __getattr__(self, args):
    if args == "Merge2": args = "Merge"
    return nuke.NodeConstructor(args)
nuke.nodes = MyCustomNodes()
```

Tip

You can also do the same using the **nuke.createNode** statement rather than the **nuke.nodes.*node-name*** statement. In this case, add the following entry instead of the entry described in step 2:

```
def createMyCustomNodes(node, knobs = "", inpanel = True):
  if node == "Merge2": node = "Merge"
  return nukeOriginalCreateNode(node = node, knobs = knobs, inpanel = inpanel)
nuke.createNode = createMyCustomNodes
```

Getting Information on the Nuke Environment You Are Running

In the Nuke module, there is an object called env. This object gives you information about the particular Nuke environment you are running. You can access it in a dictionary form: **nuke.env["key"]**.

Currently, you can use the following statements to get the following information:

- **nuke.env ["PluginExtension"]**. This returns the file extension for plug-ins. The extension can be .dll on Windows, .so on Linux, and .dylib on Mac.
- **nuke.env ["NukeVersionMajor"]**. This returns the first number in the version number of Nuke. For example, if you were running Nuke 5.0v2, this would return 5.
- **nuke.env ["NukeVersionMinor"]**. This returns the second number in the version number of Nuke. For example, if you were running Nuke 5.0v2, this would return 0.
- **nuke.env ["NukeVersionRelease"]**. This returns the number after the letter **v** in the version number of Nuke. For example, if you were running Nuke 5.0v2, this would return 2.
- **nuke.env ["NukeVersionPhase"]**. This returns the last string in the version number of a Nuke beta. For example, if you were running Nuke 5.0v2b3, this would return b.
- **nuke.env ["NukeVersionPhaseNumber"]**. This returns the last number in the version number of a Nuke beta. For example, if you were running Nuke 5.0v2b3, this would return 3.
- **nuke.env ["NukeVersionDate"]**. This returns the date the version of Nuke you are running was built.
- **nuke.env ["NukeVersionString"]**. This returns the entire version number of Nuke.
- **nuke.env ["threads"]**. This returns the number of threads that will be spawned when rendering.
- **nuke.env ["numCPUs"]**. This returns the number of detected CPUs.
- **nuke.env ["gui"]**. This returns True if you are using the graphical user interface of Nuke, and False if you are running Nuke from a terminal or command prompt.
- **nuke.env ["ExecutablePath"]**. This returns the full path to Nuke's executable.
- **nuke.env ["ple"]**. This returns True if you are running the Personal Learning Edition of Nuke, and False if you are running the commercial version.
- **nuke.env ["WIN32"]**. This returns True if you are running Nuke on Windows, and False if you are not.
- **nuke.env ["MACOS"]**. This returns True if you are running Nuke on a Mac, and False if you are not.

- **nuke.env ["64bit"].** This returns True if you are running a 64-bit Nuke.

To find out what's in the Nuke "global environment", you can also use the following statement:
print nuke.env

Some of the items are read-only (for example, '64-bit' and 'ple'), and others can be set (for example, 'threads').

Clearing Out the Current Nuke (.nk) Script

To clear out the current Nuke script without opening a new Nuke window, use the following statement:
nuke.scriptClear()

Creating Views for a Stereoscopic Project

To create a new view for a stereoscopic or multi-view project, use the following statement:
nuke.Root().addView("*name*")
where *name* is the name of the new view. For example, to add a view called left, you should use the following: **nuke.Root().addView("left")**.

Adjusting Parameter Values in Stereoscopic Projects

When you have created multiple views in your project, you can adjust node parameters separately for each view. To do so, you need to use the **setValue** statement, which takes a **view** argument.

For example, if you have created views called *left* and *right* in your project, you can do the following:

```
n = nuke.nodes.Blur()
n['size'].splitView()
n['size'].setValue(10, view = "left")
n['size'].setValue(20, view = "right")
```

This adds a Blur node into your script, splits the Size parameter off, and gives the left view the value 10 and the right view the value 20.

Automating Procedures

Okay, so you know how to use the Script Editor to type in a sequence of Python statements that take care of a procedure. But so far, you've still sat by your computer typing the statements in. It's clearly time to automate the procedure. All you need to do is save your statements, and when you want to use them again later, import them into the Script Editor.

To save statements in a Python module:

1. On the top of the Script Editor, click the **Save a script** button.
2. Save the script with the extension .py (for example **firstmodule.py**) in a directory that is contained in the sys.path variable. (To see these directories, enter **print sys.path** in the Script Editor. To add a directory to the sys.path variable, enter **sys.path.append ("*directory*")** where *directory* represents the directory you want to add.)

You have now created your first Python module.

To open a Python script in the Script Editor:

1. Click the **Load a script** button. on top of the Script Editor. The **Script to open** dialog opens.
2. Navigate to the Python module that contains the script you want to open and click **Open**.

Nuke opens the script in the input pane of the Script Editor, but does not execute it.

To import and execute a Python script:

1. On top of the Script Editor, click the **Source a script** button The **Script to open** dialog opens.
2. Navigate to the Python module that contains the script you want to import and click **Open**.

OR

In the input pane, enter:
import *module*
where ***module*** represents the name of your Python module without the file extension, for example
import firstmodule

Nuke imports the Python module and performs the procedure defined in the module.

Note

Importing the module is done according to Python's default rules. During the import, the module is searched in the following locations and order:

1. In the current directory.

2. In the directories contained in the **PYTHONPATH** environment variable, if this has been defined. (To view these directories, enter **echo $PYTHONPATH** in a command shell.)

3. In an installation-dependent default directory.

During the search, the variable **sys.path** is initialized from these directories. Modules are then searched in the directories listed by the **sys.path** variable. To see these directories, execute the statement **print sys.path** in the Script Editor.

Getting Help

In the scope of this user guide, it's not possible to go into detail with Python and all the scripts available. However, there are several sources of more information that you may find useful if you need help using Python.

More Documentation

To view documentation on Nuke's Python bindings, launch Nuke and select **Help** > **Documentation** > **Python Scripting**.

Viewing More Examples

We only described a few examples of Python scripts in this chapter, but there are more. You can find them in the following location:

- **On Windows**:
  drive letter:\Program Files\Nuke5.1v5\nukescripts or
  drive letter:\Program Files (x86)\Nuke5.1v5\nukescripts
- **On Mac OS X**:
  /Applications/Nuke5.1v5/Nuke5.1v5.app/Contents/MacOS/nukescripts
- **On Linux**:
  /usr/local/Nuke5.1v5/nukescripts

To view an example, select one of the .py files and open it in any text editor.

Using the Help Statement

Possibly the quickest way of getting help on the available scripts is to enter the following in the Script Editor:
help (nuke)

This statement lists many of the available Python commands in an alphabetical order together with their descriptions.

You can also get help on more specific things. For example, the statement
help (nuke.Knob.setValue)
would give you a description of what the **setValue (...)** method does.

Python on the Web

To read more about Python, check out its documentation, or interact with other Python users, visit the Python programming language official web site at http://www.python.org/.

Using wxPython in Nuke to Write Custom GUIs

WxPython is a collection of Python bindings for the graphical user interface (GUI) toolkit wxWidgets. Currently, Nuke works with wxPython 2.8 on Windows, Linux, and Mac.

You can use wxPython to write custom wxWidgets applications or GUIs in Python and use them within Nuke. WxPython needs to be installed separately to Nuke. You can get a copy of wxPython from http://wxpython.org/. WxPython includes wxWidgets, so you do not need to install the latter.

Both wxPython and wxWidgets are open-source software that you can download and use free of charge.

Installation

To use wxPython in Nuke, you need to install the following tools.

On Windows

1. Install Python 2.5.1 from www.python.org.
2. Install wxPython for 2.5 unicode from http://wxpython.org/. You can install it anywhere you like as long as you point the sys.path variable to it, as described in step 3. Only if you install wxPython inside *drive letter*:\Program Files\Nuke5.1v5\lib\site-packages or *drive letter*:\Program Files (x86)\Nuke5.1v5\lib\site-packages, you do not need to perform step 3.
3. Point the sys.path variable to where the wxPython modules are installed. To see the directories contained in the sys.path variable, enter **print sys.path** in the Script Editor. To add a directory to the sys.path variable, enter **sys.path.append ("directory")**. In this statement, **directory** represents the directory you want to add, for example **sys.path.append ("C:\\Python25\\Lib\\site-packages\\wx-2.8-msw-unicode")**.

On Mac OS X

1. Install Python 2.5.1 from www.python.org.
2. Install wxPython for 2.5 unicode from http://wxpython.org/.

   You can install Python and wxPython anywhere you like as long as you point the sys.path variable to them, as described in step 3.
3. Point the sys.path variable to where the Python and wxPython modules are installed. To see the directories contained in the sys.path variable, enter **print sys.path** in the Script Editor. To add a directory to the sys.path variable, enter **sys.path.append ("directory")**. In this statement, **directory** represents the directory you want to add, for example
   - **sys.path.append ("/System/Library/Frameworks/Python.framework/Versions/2.5/Extras/lib/python")**
   - **sys.path.append ("/System/Library/Frameworks/Python.framework/Versions/2.5/Extras/lib/python/wx-2.8-mac-unicode")**

On Linux

1. Install Python 2.5.1 from www.python.org.
2. Install wxPython for 2.5 unicode from http://wxpython.org/.

   You can install Python and wxPython anywhere you like as long as you point the sys.path variable to them, as described in step 3. Only if you install them inside /usr/local/Nuke5.1v5/site-packages, you do not need to perform step 3.
3. Point the sys.path variable to where the Python and wxPython modules are installed. To see the directories contained in the sys.path variable, enter **print sys.path** in the Script Editor. To add a directory to the sys.path variable, enter **sys.path.append ("directory")**. In this statement, **directory** represents the directory you want to add, for example
   - **sys.path.append ("/usr/lib/python2.5/site-packages")**
   - **sys.path.append ("/usr/lib/python2.5/site-packages/wx-2.8-gtk2-unicode")**

General Workflow

After you've installed the necessary tools, you are ready to use wxPython in Nuke. Do the following:

1. Write an application using wxPython. Save the application in a .py file in a directory that's contained in the sys.path variable, for example in $HOME/.nuke. (To see these directories,

enter **print sys.path** in the Script Editor. To add a directory to the sys.path variable, enter **sys.path.append ("directory")** where directory represents the directory you want to add.)

2. To be able to use a helper script that you'll need in the next step, enter the following in your .py file:

   from nukescripts import utils, pyWxAppUtils

3. Pass the application to the pyWxAppUtils.py helper script in the following manner:

   pyWxApp = pyWxAppUtils.pyWxAppHelper(MyApp, start = True)

   The pyWxAppUtils.py is a helper script that executes a wxPython application from a separate thread in Nuke. You can find the script in the following location:

   - **On Windows**:

     drive letter:\Program Files\Nuke5.1v5\nukescripts or

     drive letter:\Program Files (x86)\Nuke5.1v5\nukescripts

   - **On Mac OS X**:

     /Applications/Nuke5.1v5/Nuke5.1v5.app/Contents/MacOS/nukescripts

   - **On Linux**:

     /usr/local/Nuke5.1v5/nukescripts.

4. In order to communicate with Nuke's event loop, make sure any Nuke Python calls go through either **nukescripts.utils.runInMainThread** or **nukescripts.utils.runInMainThreadWithResult**. For example, to create a Blur node, you could use:

   - **nukescripts.utils.runInMainThread(nuke.nodes.Blur)**

     or if the result of the call is needed (for example, in this case to get the node object):

   - **b = nukescripts.utils.runInMainThreadWithResult(nuke.nodes.Blur)**

Note

You should only use these calls when interacting with Nuke from a separate thread. Otherwise, Nuke may hang.

Using PyQt4 in Nuke to Write Custom GUIs

PyQt is a collection of Python bindings for Qt Software's graphical user interface (GUI) toolkit Qt. PyQt is developed by Riverbank Computing, and combines the advantages of Python and Qt. Currently, Nuke works with PyQt4 on Windows and Linux but not on Mac OS X.

With PyQt, you can write custom Qt applications or GUIs in Python and use them within Nuke. Because many other applications, such as Maya and Houdini, also support PyQt, you can share the same UI across these applications without having to replicate or build it separately for each of them.

Both PyQt and Qt need to be installed separately to Nuke. You can get a copy of PyQt from www.riverbankcomputing.co.uk, and a copy of Qt from http://www.qtsoftware.com/.

Installation

To install the following tools, follow the instructions given by Qt Software and Riverbank Computing.

1. Obtain a copy of Qt from http://www.qtsoftware.com/ and install it on your machine. Qt is a C++ GUI library.
2. Download and install SIP from www.riverbankcomputing.co.uk. SIP is a tool for creating Python bindings for C++ libraries. Make sure you compile it against Python 2.5.1.
3. Download and install a copy of PyQt4 from www.riverbankcomputing.co.uk. PyQt is a collection of Python bindings for Qt.

Where you install these applications does not matter as long as the sys.path variable points to where the Python modules are installed. (To see the directories contained in the sys.path variable, enter **print sys.path** in the Script Editor. To add a directory to the sys.path variable, enter **sys.path.append ("directory")**, where **directory** represents the directory you want to add.)

General Workflow

This is the general workflow for using PyQt in Nuke:

1. Initialise PyQt from a separate thread in Nuke. An example of how to do this can be found in a file called pyQtAppUtils.py in the nukescripts folder. See *PyQtAppUtils.py: Launching PyQt from a Separate Thread in Nuke* on page 400.
2. Start a separate GUI application (that is, any application that you have written to control Nuke features) and have it run concurrently with Nuke. An example of how to do this can be found in a file called pyQtRender.py in the nukescripts folder. For more information, see *PyQtRender.py: Using a GUI built in QtDesigner* on page 400.

   To be able to use the pyQtAppUtils.py example script that initialises PyQt from a separate thread, make sure you use the following from the pyQtRender.py file in your own projects:

   from nukescripts import utils, pyQtAppUtils
3. In order to communicate with Nuke's event loop, make sure any Nuke Python calls go through either **nukescripts.utils.runInMainThread** or **nukescripts.utils.runInMainThreadWithResult**. For example, to create a Blur node, you could use:
   - **nukescripts.utils.runInMainThread(nuke.nodes.Blur)**

     or if the result of the call is needed (for example, in this case to get the node object):
   - **b = nukescripts.utils.runInMainThreadWithResult(nuke.nodes.Blur)**

Note

You should only use these calls when interacting with Nuke from a separate thread. Otherwise, Nuke may hang.

Example

In the nukescripts folder, there are example modules that illustrate how to use PyQt to create a custom GUI inside Nuke. You can find the modules in the following location:

- **On Windows**:

drive letter:\Program Files\Nuke5.1v5\nukescripts or
drive letter:\Program Files (x86)\Nuke5.1v5\nukescripts

- **On Linux**:
  /usr/local/Nuke5.1v5/nukescripts

The example consists of two modules: pyQtAppUtils.py and pyQtRender.py. PyQtAppUtils.py launches PyQt from a separate thread in Nuke, while pyQtRender.py is an example of how to use a GUI built in QtDesigner. You can run both modules by entering the following in the Script Editor:

```
from nukescripts.pyQtExamples import pyQtRender
pyQtRender.startQtRenderDialog()
```

What the two modules do is described below in more detail.

PyQtAppUtils.py: Launching PyQt from a Separate Thread in Nuke

The pyQtAppUtils.py file is an example of a module that initialises PyQt from a separate thread in Nuke. To see what the module contains, open the pyQtAppUtils.py file from the nukescripts folder in a text editor.

PyQtRender.py: Using a GUI built in QtDesigner

The pyQtRender.py module first imports the pyQtAppUtils.py module that launches PyQt from a separate thread in Nuke. It then creates a dialog with a Render button. The Render button renders the Write node that is currently selected in the dialog.

Figure 18-4: The pyQtRender.py example uses a dialog built in QtDesigner.

To see what this module contains, open the pyQtRender.py file from the nukescripts folder in a text editor.

19 Configuring Nuke

This chapter shows visual effects supervisors how to configure Nuke for multiple artists, prior to the start of a project. These are the common application settings discussed in this chapter:

- Command-line operations
- Environment variables
- Plug-in directories
- Favourite directories
- Menu and Toolbar options
- Image formats
- Gizmos (Nuke group nodes or subscripts that allow only select modifications)
- Custom plug-ins (binary plug-ins made via the Nuke software developers kit)
- Generic TCL ("TiCkLe") scripts
- Template scripts
- Common preferences
- Script's lookup tables (LUTs)

What Is a Terminal and How Do I Use One?

Many tasks in this chapter tell you to enter commands from a terminal or shell. This refers to a window where you can enter commands directly rather than making selections through a user interface.

The following describes how to open such a window for your operating system.

- Linux: Click the right mouse button over the desktop and choose **New Terminal** (or **Open Terminal**) from the pop-up menu.
- Windows: From the **Start** menu, choose **All Programs > Accessories > Command Prompt**.
- Mac OS X: Click on the **Terminal** dock icon.
  OR
  Browse to the **Applications > Utilities** folder on your system hard drive, and double-click the **Terminal** icon.

Inside the terminal or shell, you'll see a command prompt, which looks similar to this:

Figure 19-1: Command prompt window for Windows XP.

Once you see the command prompt, you can enter commands to perform various tasks like listing the files in a directory or running programs. Here are some specific examples:

- On Linux or Mac OS X, type **pwd** and press **Enter** to view the path of the current directory. On Windows, the equivalent command would be **cd**.
- On Linux or Mac OS X, type **ls** and press **Enter** to view a list of files in the current directory. On Windows, the equivalent command would be **dir**.
- On Linux, Mac OS X and Windows, type **cd** followed by a full pathname and press **Enter** to change directories.

Command Line Operations

Command-line flags activate various options when you launch Nuke from a shell, and provide additional functionality to Nuke. First let's discuss how to launch Nuke from a shell.

On Mac OS X

Open a Terminal and change directory as follows:

```
cd /Applications/Nuke5.1v5/Nuke5.1v5.app/
```

To launch Nuke type this command:

```
./Nuke5.1v5
```

Alternatively you can set an alias to point to Nuke and then you can launch Nuke from any directory.

```
alias nuke /Applications/Nuke5.1v5/Nuke5.1v5.app/Nuke5.1v5
```

Change to your HOME directory:

```
cd
```

Launch nuke:

```
nuke
```

Tip

You can add aliases to a .cshrc file in your home directory so that they are activated each time you open a shell. See your Systems Administrator for help setting this up.

Now you can start experimenting with command line flags on launching Nuke. Here's one that displays the version number and build date.

```
nuke -version
```

If you have a Nuke script, you can render it on the command line without opening the GUI version. Here's an example that renders a hundred frames of a Nuke script:

```
nuke -x myscript.nk 1,100
```

To display a list of command line flags (switches) available to you, use the following command:

```
nuke -help
```

Here's that list of command line flags in a table:

Switch/Flag	Action
-b	Background mode. This launches Nuke and returns control to the terminal, so you get your prompt back. This is equivalent to appending a command with an & to run in the background.
-c *size* (**k**, **M**, *or* **G**)	Limit the cache memory usage, where *size* equals a number in bytes. You can specify a different unit by appending **k** (kilobytes), **M** (megabytes), or **G** (gigabytes) after *size*.
-d <x server *name*>	This allows Nuke to be viewed on one machine while run on another. (Linux only and requires some setting up to allow remote access to the X Server on the target machine).
-f	Open Nuke script at full resolution. Scripts that have been saved displaying proxy images can be opened to show the full resolution image using this flag. See also -p.
-h	Display command line help.
-help	Display command line help.
-i	Use an interactive (nuke_i) FLEXlm license key. This flag is used in conjunction with background rendering scripts using -x. By default -x will use a nuke_r license key, but -ix will background render using a nuke_i license key.
-l	New read or write nodes will have the colourspace set to linear rather than default.
-m *#*	Set the number of threads to the value specified by *#*.
-n	Open script without postage stamps on nodes.

Switch/Flag	Action
-p	Open Nuke script at proxy resolution. Scripts that have been saved displaying full resolution images can be opened to show the proxy resolution image using this flag. See also -f.
-q	Quiet mode. This stops all printing to the shell.
-s *#*	Sets the minimum stack size for each thread in bytes. This defaults to 16777216 (16 MB). The smallest allowed value is 1048576 (1 MB).
-t	Terminal mode. This allows you to enter Python commands without launching the GUI. A *>>>* command prompt is displayed during this mode. Enter quit() to exit this mode and return to the shell prompt. This mode uses a nuke_r license key by default, but you can get it to use a nuke_i key by using the -ti flag combo.
-V	Verbose mode. In the terminal, you'll see explicit commands as each action is performed in Nuke.
-v	This command displays an image file inside a Nuke Viewer. Here's an example: nuke -v image.tif
-version	Display the version information in the shell.
-x	eXecute mode. Takes a Nuke script and renders all active Write nodes. Note that this mode uses a FLEXlm nuke_r license key. To use a nuke_i license key, use -xi. This is the syntax: nuke -x myscript.nk On Windows, you can press **Ctrl**+**Break** to cancel a render without exiting if a render is active, or exit if not. **Ctrl**/**Cmd**+**C** exits immediately. On Mac and Linux, **Ctrl**/**Cmd**+**C** always exits.
-X *node*	Render only the Write node specified by *node.*
--	End switches, allowing script to start with a dash or be just - to read from stdin

General Syntax

This is the general syntax for using these options when launching Nuke at the command prompt:
nuke *<switches> <script> <argv> <ranges>*

<switches> - modifies the behaviour of Nuke when run from the command line. A list of switches is given in the table above. These are sometimes called flags.

<script> - the name of the Nuke script.

<argv> - an optional argument that can be used in Nuke. See the example below.

<ranges> - this is the frame range you want rendering.

Examples

Let's consider some practical examples.

To launch Nuke and open a script.

```
nuke myscript.nk
```

Crazy I know, but I've called my script, -myscript.nk, and the hyphen at the start of the filename has confused Nuke. To get round this if you don't want to rename your file use the double hyphen syntax:

```
nuke -- -myscript.nk
```

Viewing Images

To display an image:

```
nuke -v polarbear.tif
```

To display several images:

```
nuke -v polarbear.tif whiteflower.psd mountains.cin
```

To display an image sequence (taxi.0001.tif, taxi.0002.tif,...,taxi.0050.tif):

```
nuke -v taxi.%04d.tif 1,50
```

Rendering on the command line

To render frame 5 of a Nuke script:

```
nuke -x myscript.nk 5
```

To render frames 30 to 50 of a Nuke script:

```
nuke -x myscript.nk 30,50
```

To render every tenth frame of a 50 frame sequence of a Nuke script:

```
nuke -x myscript.nk 1,50,10
```

This will render frames 1, 11, 21, 31, 41.

In a script with two write nodes called WriteBlur and WriteInvert this command will just render frames 1 to 20 from the WriteBlur node:

```
nuke -X WriteBlur myscript.nk 1,20
```

Using [argv 0]

Let's use [argv] to vary the output file name. Launch the GUI version of Nuke and create a node tree that puts a checker into a Write node. Open the write node property panel by double clicking on it and in the file text field enter this filename:
[argv 0].%04d.tif
Save the script and quit Nuke. On the command line type:

```
nuke -x myscript.nk mychecker 1,5
```

This will render 5 frames (mychecker.0001.tif, mychecker.0002.tif, etc.).

You can add another variable to control the output image file type. The file text field needs this:
[argv 0].%04d.[argv 1]
and then render the script using this command:

```
nuke -x myscript.nk mychecker cin 1,5
```

to get mychecker.0001.cin, mychecker.0002.cin, etc.

The <argv> string can be any [argv n] expression to provide variable arguments to the script. These must be placed between the *<script>* and the *<ranges>* on the command line. You can include multiple expressions, but each must begin with a non-numeric character to avoid confusion with the frame range control. For more information on expressions, see Chapter 16: *Expressions* on page 357.

Using Python to convert TIFFs to JPGs

This command line method will convert 5 tiff frames to jpeg.

```
nuke -t
>>> r=nuke.nodes.Read(file="myimage.%04d.tif")
>>> w=nuke.nodes.Write(file="myimage.%04d.jpg")
>>> nuke.execute("Write1", 1,5)
>>> quit()
```

It's a bit tedious typing these commands in line by line. So let's put them in a text file called imageconvert.py and get Nuke to execute the Python script.

```
cat imageconvert.py
r=nuke.nodes.Read(file="myimage.%04d.tif")
w=nuke.nodes.Write(file="myimage.%04d.jpg")
nuke.execute("Write1", 1,5)

nuke -t < imageconvert.py
```

Environment Variables

Environment variables are named variables used to store a value, such as a specific file path. They can be used to influence Nuke's behaviour. For example, Nuke uses the information stored in them to define where to place certain files.

Setting Environment Variables

To set an environment variable:

On Windows

1. Right-click on **My Computer** and select **Properties**.
2. Go to the **Advanced** tab.
3. Click the **Environment Variables** button. The *Environment Variables* dialog opens.

4. Click the **New** button under either **User variables** or **System variables**, depending on whether you want to set the variable for the current user or all users. To set environment variables for all users, you need to have administrator priviledges.
5. In the **Variable name** field, enter the name of the environment variable you want to set. For a list of the environment variables that Nuke understands, see *Nuke Environment Variables* on page 409.
6. In the **Variable value** field, enter the value for the variable. The value can be a directory path, for example.
7. Click **OK**.

Note

When editing existing system variables or adding or deleting either user or system variables on Windows Vista, you need to log off and on again before your changes to environment variables take effect.

On Mac

On Mac OS X, there is a special environment file - a .plist or property list file - which is read every time a user logs in. You may need to create the .plist file if it doesn't already exist in ~/.MacOSX/environment.plist (where "~" indicates the user's home directory, and "." a hidden directory).
Environment variables set using the .plist file are read both when Nuke is launched from the Nuke icon and when it's launched from the Terminal.

1. Open a Terminal window. By default, you should be in your home directory (your own directory in the Users folder). Enter **pwd** (present working directory) to verify this.
2. Enter **ls -al** to see a list of files in that directory. If .MacOSX is not in the list, enter **mkdir .MacOSX** to create the directory.
3. To create your .plist file, launch TextEdit.
4. Copy the following into the document and edit the "key" and "string" entries:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>NUKE_PATH</key>
    <string>/SharedDisk/Nuke</string>
    <key>OFX_PLUGIN_PATH</key>
    <string>/SharedDisk/OFX</string>
</dict>
</plist>
```

   This example sets two environment variables: NUKE_PATH and OFX_PLUGIN_PATH. NUKE_PATH points to /SharedDisk/Nuke/, and OFX_PLUGIN_PATH to /SharedDisk/OFX.

   For a list of the environment variables that Nuke understands, see *Nuke Environment Variables* on page 409.
5. Once you are happy with the document, select **Format** > **Make Plain Text**.

6. Save the file to your home directory with the name **environment.plist**. Make sure a .txt extension is not added to the end of the file name.
7. Quit TextEdit, and launch a Terminal window. Enter **pwd** to make sure you are in your home directory.
8. To move the environment.plist file from your home directory into the .MacOSX directory, enter **`mv environment.plist .MacOSX`**.
9. Log out and log in again.

On Linux

1. The procedure for setting an environment variable depends on what your default shell is. To get the name of the shell you are using, launch a shell and enter **`echo $SHELL`**.
2. Depending on the output of the previous step, do one of the following:
   - If your shell is a csh or tcsh shell, add the following command to the .cshrc or .tcshrc file in your home directory: **`setenv VARIABLE value`**. Replace **`VARIABLE`** with the name of the environment variable and **`value`** with the value you want to give it, for example **`setenv NUKE_PATH /SharedDisk/Nuke`**.
   - If your shell is a bash or ksh shell, add the following command to the .bashrc or .kshrc file in your home directory: **`export VARIABLE=value`**. Replace **`VARIABLE`** with the name of the environment variable and **`value`** with the value you want to give it, for example **`export NUKE_PATH=/SharedDisk/Nuke`**.

   For a list of the environment variables that Nuke understands, see *Nuke Environment Variables* on page 409.

To check if an environment variable exists:

On Windows

1. Select **Start** > **All Programs** > **Accessories** > **Command Prompt**.
2. In the command window that opens, enter **`echo %VARIABLE%`**. Replace **`VARIABLE`** with the name of the environment variable. For example, to check if NUKE_DISK_CACHE is set, enter **`echo %NUKE_DISK_CACHE%`**.

If the variable is set, its value is displayed in the command window.

On Mac or Linux

1. Launch Terminal or a shell.
2. Enter **`echo $VARIABLE`**. Replace **`VARIABLE`** with the name of the environment variable. For example, to check if NUKE_DISK_CACHE is set, enter **`echo $NUKE_DISK_CACHE`**.

If the variable is set, its value is displayed in the Terminal or shell window.

To display a list of set environment variables:

On Windows

1. Select **Start** > **All Programs** > **Accessories** > **Command Prompt**.
2. In the command window that opens, enter **`set`**.

A list of all the environment variables that are set is displayed in the command window.

On Mac or Linux

1. Launch Terminal or a shell.
2. Enter **`printenv`**.

A list of all the environment variables that are set is displayed in the Terminal or shell window.

Nuke Environment Variables

The following table lists the environment variables Nuke recognises.

Environment Variable	Description
FOUNDRY_LICENSE_FILE	The location of the Nuke license file (a plain text file called nuke.lic), if the following recommended location is not used: **On Mac OS X and Linux**: /usr/local/foundry/FLEXlm **On Windows XP**: *drive letter*:\Program Files\The Foundry\FLEXlm **OR** *drive letter*:\Program Files (x86)\The Foundry\FLEXlm (for 32-bit Nuke on 64-bit Windows) **On Windows Vista:** *drive letter*:\ProgramData\The Foundry\FLEXlm
NUKE_PATH	The location where files related to Nuke customisations are stored. For more information, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.
OFX_PLUGIN_PATH	The location where Nuke looks for OFX plug-ins. For more information, see *Loading OFX Plug-ins* on page 411.
NUKE_TEMP_DIR	The location where Nuke saves any temporary files that do not have a particular place defined for them. This is also where the **flipbook cache** setting in the Preferences defaults to.
NUKE_DISK_CACHE	The location where Nuke saves all recent images displayed in the Viewer. Ideally, this should be a local disk with the fastest access time available. It should also have enough space for the maximum flipbook cache size. If this variable is not set, the location is determined by the **flipbook cache** setting in the Preferences. By default, this setting points to NUKE_TEMP_DIR.
NUKE_DISK_CACHE_GB	The maximum size the flipbook cache can reach (in gigabytes). If this variable is not set, the location is determined by the **flipbook cache size** setting in the Preferences.

Environment Variable	Description
FC_PATH	If you want to use a different version of FrameCycler than the one shipped with Nuke, you can use this to set the base directory of FrameCycler. This points to the FrameCycler binary.
FC_HOME	This points to the directory that contains bin/framecycler.
FC_DISK_CACHE	The location where Nuke saves all recent images flipbooked with FrameCycler. If this variable is not set, the location is determined by NUKE_DISK_CACHE.

Loading Gizmos, Plug-ins, and TCL scripts

On start-up, Nuke scans various directories for files that customise the behaviour of Nuke. It looks for information on favourite directories, menu options, image formats, gizmos, NDK plug-ins, generic TCL scripts, and preferences.

It looks in specific subdirectories of the user's home directory and the Nuke application directory as follows:

- **Linux:**
  /users/*login name*/.nuke
  /usr/local/Nuke5.1v5/plugins
- **Mac OS X:**
  /Users/*login name*/.nuke
  /Applications/Nuke5.1v5/Nuke5.1v5.app/Contents/MacOS/plugins
- **Windows:**
  In the .nuke directory, which can be found under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable - which is generally of the form *drive letter*:\Documents and Settings*login name*\ (Windows XP) or *drive letter*:\Users*login name*\ (Windows Vista).

  To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you will get an error.

  drive letter:\Program Files\Nuke5.1v5\plugins (or, when using 32-bit Nuke on 64-bit Windows, *drive letter*:\Program Files (x86)\Nuke5.1v5\plugins)

If you want Nuke to look for plug-ins somewhere else rather than in these default locations, you can also define a common plug-in path yourself. Thus, by defining the Nuke plug-in path, you can assign yourself a common shared directory from which to control Nuke for multiple artists.

To define the Nuke plug-in path:

1. On each artist's machine, create an environment variable called **NUKE_PATH**.

2. Assign the **NUKE_PATH** environment variable to the pathname of the directory where you intend store files related to Nuke customisations.

For example, on Mac OS X:

```
setenv NUKE_PATH /SharedDisk/Nuke
```

Loading Python Scripts

On start-up, Nuke scans various directories for Python scripts that customise the behaviour of Nuke. For example, the buttons on the toolbars are configured with the toolbars.py script.

It looks in specific subdirectories of the Nuke application directory as follows:

- **Linux:**
  /usr/local/Nuke5.1v5/nukescripts
- **Windows:**
  drive letter:\Program Files\Nuke5.1v5\nukescripts or
  drive letter:\Program Files (x86)\Nuke5.1v5\nukescripts
- **Mac OS X:**
  /Applications/Nuke5.1v5/nukescripts

Warning

It's worth saying that you should edit these files with care as mistakes could stop Nuke from running.

Loading OFX Plug-ins

On start-up, Nuke scans various directories for OFX plug-ins that will bring additional functionality to Nuke. Paths to these directories vary between operating systems, but here are examples of where you may find them:

- **Linux:**
  /usr/OFX/Nuke
  /usr/OFX/Plugins
- **Windows:**
  C:\Program Files\Common Files\OFX\Nuke (or, when using 64-bit Nuke on 64-bit Windows, \Program Files (x86)\Common Files\OFX\Nuke)
  C:\Program Files\Common Files\OFX\Plugins (or, when using 64-bit Nuke on 64-bit Windows, \Program Files (x86)\Common Files\OFX\Plugins)
- **Mac OS X:**
  /Library/OFX/Nuke
  /Library/OFX/Plugins

If you want Nuke to look for OFX plug-ins somewhere else you can. Just define the environment variable OFX_PLUGIN_PATH to point to the new directory.

For example, on Mac OS X:

```
setenv OFX_PLUGIN_PATH /SharedDisk/OFX
```

Defining Common Favourite Directories

Favourite directories can be accessed by artists with a single click from any Nuke file browser. Typically you would create these favourites for common directories on a project.

To define a common set of favourite directories:

1. Create a file called **menu.py** in your plug-in path directory.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Add an entry in the following format:

   nuke.addFavoriteDir('DisplayName', 'Pathname')

   - Replace **DisplayName** with the string you want as the display name for the directory link, for example **'Home'** or **'Desktop'**.
   - Replace **Pathname** with the pathname to the directory to which the favourite should point.

3. In the above entry, you can also add the following optional arguments after **'Pathname'**:

   - **type**. This is an integer and a bitwise operator, OR a combination of **nuke.IMAGE**, **nuke.SCRIPT** or **nuke.FONT**.

     nuke.IMAGE restricts the favourite directory to appear only in the image file browser, which Nuke opens for file Read/Write operations.

     nuke.SCRIPT restricts the favourite directory to appear in the script file browser, which appears when you choose **File** > **Open** or a similar menu selection to open or import script files.

     nuke.FONT restricts the favourite directory to appear in the font browser.

   - **icon='Name'.** Replace **Name** with the name and file extension of the png (or xpm) image you wish to use as the icon for the favourite directory. This image must be stored in your Nuke plug-in path directory. It should be 24 x 24 pixels in size.
   - **tooltip='My tooltip'**. Replace **My tooltip** with the string you wish to display as pop-up help.

Example 1

The following entry would create a favourite called **DisasterFlickStore** which appears on all File Browsers invoked from Read nodes and which points to the **/job/DisasterFlick/img** directory.

```
nuke.addFavoriteDir ('DisasterFlickStore', '/job/DisasterFlick/img',
nuke.IMAGE)
```

Figure 19-2: The result of example 1.

Example 2

The following entry would create a favourite called **Test**. It appears on all File Browsers invoked from Read nodes or by selecting **File** > **Open** and points to the **/job/Test** directory. The entry also defines *Test Images and Scripts* as the tool tip for the favourite directory.

```
nuke.addFavoriteDir ('Test', '/job/Test', nuke.IMAGE | nuke.SCRIPT,
tooltip='Test images and Scripts')
```

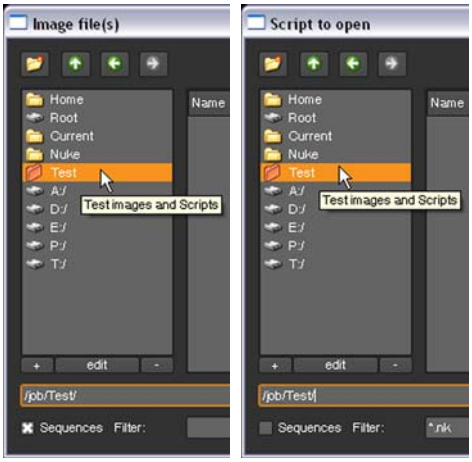


Figure 19-3: The result of example 2.

Handling File Paths Cross Platform

If your facility uses Nuke on several operating systems, you may want to configure Nuke to replace the beginnings of file paths so that scripts created on one platform will also work on another.

For example, to ensure file paths created on Windows also work on Linux and vice versa, you can do the following:

1. Create a file called **init.py** in your plug-in path directory if one doesn't already exist.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Open the init.py file in a text editor and add an entry in the following format:

```
import platform


def filenameFix(filename):
if platform.system() in ("Windows", "Microsoft"):
   return filename.replace( "/SharedDisk/", "p:\" )
else:
  return filename.replace( "p:\", "/SharedDisk/" )
return filename
```

   This way, the Windows file paths (beginning with *p:* in the above example) will be replaced with the Linux file paths (beginning with */SharedDisk/*) whenever a Nuke script is used on Linux. Otherwise, the Windows file paths are used.

Defining Custom Menus and Toolbars

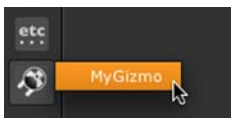
You can freely add custom menus and menu options as well as toolbars and toolbar options to the Nuke interface. Artists can then use these options to trigger gizmos and plug-ins stored in the plug-in path directory.

For example, to add a new menu in the default Toolbar with an option to trigger a gizmo called MyGizmo, you can do the following:

1. In your home directory, create a directory called .nuke (if it doesn't exist already). For more information on this directory, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.
2. In the .nuke directory, create a file called **menu.py** if one does not already exist.
3. In a text editor, modify the file **menu.py**, adding the lines:

```
toolbar = nuke.toolbar("Nodes")
toolbar.addCommand( "Test/MyGizmo", "nuke.createNode('MyGizmo')")
```

   This adds a menu labeled "Test" to the default Nodes Toolbar with an item labeled "MyGizmo" that creates an instance of the node MyGizmo.



It's also possible to add items to other menus in Nuke and even create your own toolbars. The following sections cover these possibilities in detail.

To add a toolbar:

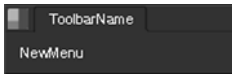
1. Create a file called **menu.py** in your plug-in path directory if one doesn't already exist.

For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Open the menu.py file in a text editor and add an entry in the following format:

```
t=nuke.toolbar("ToolbarName")
t.addCommand("NewMenu", "PythonCode", "Shortcut", icon="IconName")
```

   - Replace **ToolbarName** with the name you want to give to the toolbar. This name will appear in the content menus under "Toolbars" and above the toolbar on the title tab.
   - Replace **NewMenu** with the name of the menu you want to add to the toolbar. The following image illustrates where whatever you use to replace **ToolbarName** and **NewMenu** will appear in the new toolbar.



   - Replace **PythonCode** with relevant Python code (usually **nuke.createNode**), and, if necessary, use the name of the gizmo, generic Python script, or plug-in file you want the menu option to invoke. For ease of use, place all such referenced files inside the plug-in path directory.
   - Replace **Shortcut** with a keyboard shortcut, for example **Alt+A**, **Ctrl/Cmd+A**, or **Shift+A**. The letter **a** alone represents lower-case **a**. **F1** represents function key 1. You can combine the **Shift**, **Ctrl/Cmd**, and **Alt** keys as necessary. If you like, you can also use **#A** to represent **Alt+A,** **^A** to represent **Ctrl/Cmd+A**, and **+A** to represent **Shift+A**.
   - Replace **IconName** with the name of the png (or xpm) image you wish to use as the menu icon. This image must be stored in your Nuke plug-in path directory. It should be 24 x 24 pixels in size.

3. In the above entry, you can also add the following optional arguments in the parenthesis after **"ToolbarName"**:
   - **True**. This is the default. When True, nuke.toolbar() calls the toolbar with the given name or creates it if it does not exist. For example, **t=nuke.toolbar("Extras", True)** would either call an existing toolbar called Extras or create one if it did not already exist.
   - **False**. When False, the toolbar is not created if it does not already exist and nuke.toolbar() returns **None**. You can use this to find out if a toolbar with a given name already exists. For example, **t=nuke.toolbar("Extras", False)** would either call an existing toolbar called Extras or return None if such a toolbar did not exist.

The new toolbar does not appear by default, but is listed under **Toolbars** in the content menus. From there, you can insert it in any pane. Once you are happy with the new toolbar and its position, save the layout (select **Layout** > **Save Layout 1**). Thereafter, the toolbar will appear whenever Nuke is launched.

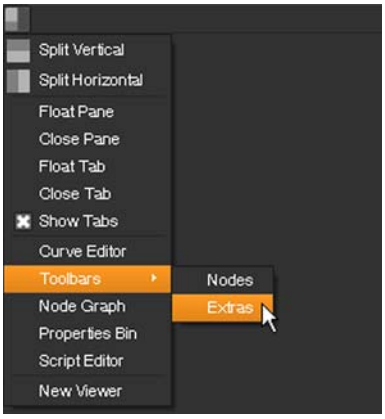


Figure 19-4: Custom toolbars are listed under "Toolbars" in the content menus.

You can build several toolbars for different tasks and save layouts that have one or another present for easy context switching.

Example 1

The following entry creates a new toolbar called **Extras**. The toolbar includes an option called **Create VectorBlur** that creates a VectorBlur node. The entry also defines **v** as the keyboard shortcut for the VectorBlur node.

```
t=nuke.toolbar("Extras")
t.addCommand("Create VectorBlur", "nuke.createNode ('VectorBlur')",
"v")
```

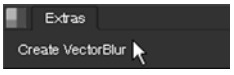
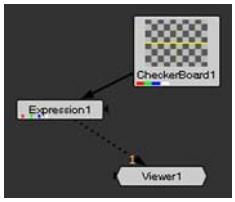


Figure 19-5: The result of example 1.

Example 2

In this example, we add an option called Autoplace to the toolbar created in example 1. This option places the selected nodes neatly one after another, as illustrated in the following images:



Before Autoplace

After Autoplace

Figure 19-6: Using Autoplace to tidy up the Node Graph.

The following entry adds the Autoplace option. It also defines **Alt+A** as the keyboard shortcut for this option.

```
def _autoplace():
  n = nuke.selectedNodes()
```

```
  for i in n:
    nuke.autoplace(i)

t=nuke.toolbar("Extras")
t.addCommand("Auto&place", "_autoplace()", "Alt+a")
```

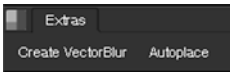


Figure 19-7: The result of example 2.

To define a menu or toolbar option:

1. If you haven't already done so, create a file called **menu.py** in your plug-in path directory. For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.
2. Open the menu.py file in a text editor and add an entry in the following format:

   ```
   menubar=nuke.menu("MenuType")
   m=menubar.addMenu("&NewMenu")
   m.addCommand("&NewItem", "PythonCode", "Shortcut", icon="IconName", index=#)
   ```

   - Replace **MenuType** with the type of menu or toolbar you want to add an item to:

     Nuke adds an item to the application main menu bar.

     Animation adds an item to the pop-up menu on the Animation button of all panels, and to the right-click pop-up menu of the Curve editor.

     Properties adds an item to the right-click menus of properties panels.

     Node Graph adds an item to the right-click menu of the Node Graph.

     Nodes adds an item to the default Toolbar.

     Viewer adds an item to the right-click menu of the Viewer.

   - Replace **NewMenu** with the menu name. Using an existing menu name appends any new options to the existing menu. You can also add options to the default Menu bar and Toolbar.
   - Replace **NewItem** with the underlying item you want to add to the menu. You may precede any character with an **&** in order to flag it as hot key trigger.
   - Replace **PythonCode** with relevant Python code (usually **nuke.createNode**) and, if necessary, use the name of the gizmo, generic Python script, or plug-in file you want

the menu option to invoke. For ease of use, place all such referenced files inside the plug-in path directory.

For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

- Replace **Shortcut** with a keyboard shortcut, for example **Alt+A**, **Ctrl/Cmd+A**, or **Shift+A**. The letter **a** alone represents lower-case **a**. **F1** represents function key 1. You can combine the **Shift**, **Ctrl/Cmd**, and **Alt** keys as necessary. If you like, you can also use **#A** to represent **Alt+A,** **^A** to represent **Ctrl/Cmd+A**, and **+A** to represent **Shift+A**.

Note

By assigning a keyboard shortcut, you can overwrite existing shortcuts. For example, if you assign the shortcut **Ctrl/Cmd+O** to a new menu item, it will no longer be used for it's default purpose which is opening a file. However, shortcuts are only overwritten in the main menu bar, the Toolbar, any user-created toolbars, and the menu you are adding the new menu item to. This means you can add a shortcut into the Node Graph, for example, without resetting the same shortcut in the Viewer. However, you cannot add a shortcut into the Node Graph without resetting the same shortcut in the main menu bar or the Toolbar.

- Replace **IconName** with the name of the png (or xpm) image you wish to use as the menu icon. This image must be stored in your Nuke plug-in path directory. It should be 24 x 24 pixels in size.
- Replace # with a number that represents the position of the item in the menu or toolbar. If you do not use an index keyword, the item will be added in the end of the menu or toolbar.

Tip

You can also put the menu name in the addCommand call, like this:

```
nuke.menu("MenuType").addCommand("NewMenu/NewItem",
"PythonCode("name")")
```

Example 1

The following entry creates a new menu and option called **Custom** > **Cue Render** in the menu bar. It inserts a gizmo called "cue_render." The entry also defines **Ctrl+R** as the keyboard shortcut for the gizmo.

```
menubar=nuke.menu("Nuke")
m=menubar.addMenu("&Custom")
m.addCommand("&Cue Render", "nuke.createNode('cue_render')", "Ctrl+R")
```

Example 2

For information on how to create a new menu in the default Toolbar with a menu item that triggers a gizmo, see the example on page 414.

Defining Common Image Formats

You may wish to define image formats (image resolutions and corresponding pixel aspect ratios) for a given project. These will appear as pulldown options on Read and Reformat nodes.

To define an image format:

1. If you haven't already done so, create a file called **menu.py** in your plug-in path directory.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Add an entry in the following format:

   ```
   nuke.addFormat(" ImageWidth ImageHeight LowerLeftCorner
   LowerRightCorner UpperRightCorner UpperLeftCorner PixelAspectRatio
   DisplayName ")
   ```

   - Replace **ImageWidth** with the width (in pixels) of the image format.
   - Replace **ImageHeight** with the height (in pixels) of the image format.
   - If you wish to define an image area that is smaller than the format resolution, replace **LowerLeftCorner, LowerRightCorner, UppperRightCorner, UpperLeftCorner** with the proper x and y coordinates (in pixels) for the lower left, lower right, upper right, and upper left corners of this smaller image area (optional).
   - Replace **DisplayName** with display name of the format. This will appear on all relevant pulldown menus.

Example 1

The following entry would create a new format called **full_aperature_anamorphic** with a resolution of 2048 by 1556 and a pixel aspect ratio of 2.0. (As the image corners are not explicitly defined, the image will cover the entirety of the format area.)

```
nuke.addFormat (" 2048 1556 2.0 full_aperture_anamorphic ")
```

Gizmos, Custom Plug-ins, and Generic TCL Scripts

Nuke allows artists and technical directors to create *gizmos*, which are simply groups of Nuke nodes that may be reused by other artists. These are equivalent to Shake's macros. Studios commonly use gizmos to consistently apply certain colour grading techniques, process incoming footage according to a particular conversion formula, and process outgoing footage in preparation for film printing.

A gizmo is a Group Node that you create and save in a separate .gizmo file in your Nuke plug-in folder. Nuke scripts can use this gizmo just like any other node type. Saved scripts only contain the name and control settings for the gizmo; the definition is in the gizmo file and it is

read at the same time the script is loaded into Nuke. Thus, you can alter the implementation of the gizmo and change all the scripts that are using it.

Note

Unlike other nodes, gizmos cannot be cloned. For more information on cloning nodes, see *Cloning Nodes* on page 38.

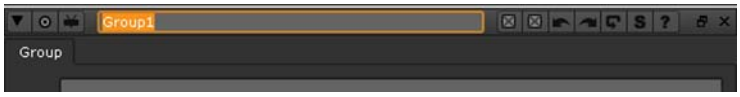
Creating and Sourcing Gizmos

Using Nuke's **Export gizmo** command, you can export a group of nodes and explicitly control which controls may be edited by the artists to ensure the processes within the gizmo are consistently applied.

Creating Gizmos

To create a gizmo:

1. Select the nodes you want to include in the gizmo.
2. Select **Other** > **Group** from the Toolbar (or press **Ctrl/Cmd+G**) to group the nodes.
3. You may want to rename the group by entering a new name in the Group properties panel title field. This step is optional, and has no effect on the saved gizmo. However, it is often a good idea to give the Group the same name you intend to use for the gizmo.



4. To control which controls the artists can adjust, follow the instructions under *Managing the Gizmo Controls* below.
5. Click the **export as gizmo** button.
6. In the file browser that appears, click **Home**. Type **.nuke/** after the path displayed at the bottom of the file browser.
7. Enter a name after the path, and append a **.gizmo** extension after the name. This is the name of the command that will be written to any saved script that's using the gizmo. The name should begin with a capital letter, because Nuke uses this as an indication that the command is a node or a gizmo. Thus, it will produce a more useful error message if the gizmo file is not found when the script is loaded.
8. Click **Save**.

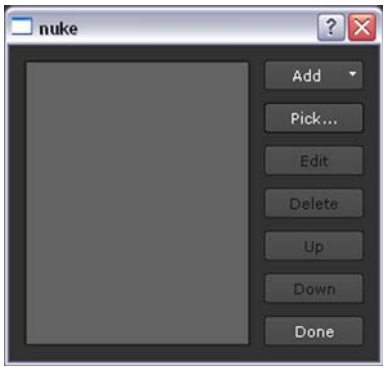
Managing the Gizmo Controls

You can add controls to your gizmo (which at this point is just a Group node) in two different ways:

- by picking and editing a control from the controls that by default exist for the nodes inside your Group node. For example, if the Group node contains a Grade node, you can add any of the Grade node controls to your gizmo properties panel.
- by adding a control you have created yourself to your gizmo properties panel.

To pick existing controls:

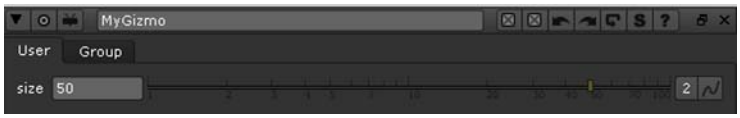
1. Right-click on the dark grey background of the Group properties panel and select **Manage User Knobs**. The following dialog opens.



2. To pick a control you want to give the users control of, click on the **Pick** button. This opens a dialog that lists all the nodes that the group contains.

Expand the list items as necessary to see the controls you can include in the gizmo controls. Select a control and click **OK**. You can also select multiple controls by **Ctrl**/**Cmd**+clicking on them, or pick a range of controls by **Shift**+clicking.

At this point, a new tab called **User** appears in the Group properties panel. The control you selected has been added to this tab. In our example, we selected the **size** control of the Text node.



The control has also been added to the Manage User Knobs dialog, ready for you to edit.

3. To edit the control you added, open the Manage User Knobs dialog and select the control from the list. Click **Edit**.

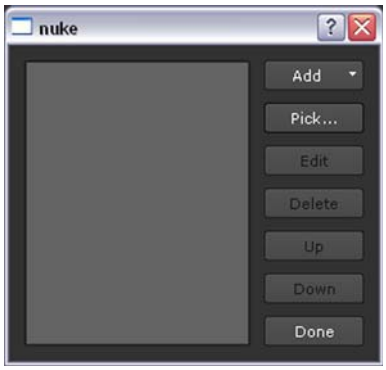
In most cases, you can edit the following:

- **Name** - Give the new control a unique name here. You need to use this name whenever you want to reference the control from scripts or via expressions. The name can only contain letters and digits. Spaces or punctuation are not allowed. This field cannot be left empty.
- **Label** - Whatever you enter here appears to the left of the control in the gizmo properties panel (or, in the case of buttons, on the button). If you leave this empty, whatever is in the **Name** field is also used as the label.
- **Tooltip** - Enter a short help text here. It will appear, along with the Name, in a pop-up tool tip when the user points the mouse at the control. If you do not provide a tool tip, whatever is in the **Name** field will be used as the tool tip.
- **Hide** - Check this to hide the control from the users. This can be useful if you want to make a new control to contain a complex expression that you can then refer to repeatedly by other controls.
- **Start new line** - Uncheck this if you want the control to appear on the same line as the previous control in the gizmo properties panel.

4. If necessary, repeat the previous three steps to add more controls to your Group node (future gizmo).
5. In the Group node properties panel, the controls are listed in the same order as they are in the Manage User Knobs dialog. To move a control up or down in the properties panel, select it in the dialog and use the **Up** and **Down** buttons.
6. Once you have all the controls you need, proceed to step 5 under *Creating Gizmos* on page 420.

To create new controls:

1. Right-click on the dark grey background of the Group properties panel and select **Manage User Knobs**. The following dialog opens.



2. To add a new control, tab, static text, or divider line to the Group (gizmo) controls, click **Add** on the Manage User Knobs dialog and select the option you want to add. This opens a dialog where you can edit the control, tab or static text you added. In most cases, you can edit the following:
   - **Name** - Give the new control a unique name here. You need to use this name whenever you want to reference the control from scripts or via expressions. The name can only contain letters and digits. Spaces or punctuation are not allowed. This field cannot be left empty.
   - **Label** - Whatever you enter here appears to the left of the control in the gizmo properties panel. If you leave this empty, whatever is in the **Name** field is also used as the label.
   - **Tooltip** - Enter a short help text here. It will appear, along with the Name, in a pop-up tool tip when the user points the mouse at the control. If you do not provide a tool tip, whatever is in the **Name** field will be used as the tool tip.
   - **Hide** - Check this to hide the control from the users. This can be useful if you want to make a new control to contain a complex expression that you can then refer to repeatedly by other controls.
   - **Start new line** - Uncheck this if you want the control to appear on the same line as the previous control in the Group properties panel.
3. Use an expression to link the control you just created to a node and its control inside the Group node. This is important, because for the new control to do anything, you need to refer to it using an expression in some other control on a node inside the Group. For more information, see the examples below and refer to *Expressions* on page 357.
4. If necessary, repeat the previous four steps to add more controls to your Group node (future gizmo).
5. In the Group node properties panel, the controls are listed in the same order as they are in the Manage User Knobs dialog. To move a control up or down in the properties panel, select it in the dialog and use the **Up** and **Down** buttons.
6. Once you have all the controls you need, proceed to step 5 under *Creating Gizmos* on page 420.

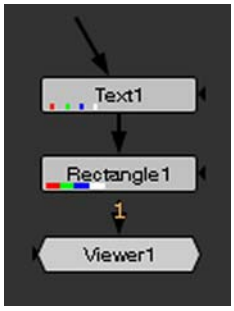
To delete controls:

1. Right-click on the dark grey background of the Group properties panel and select **Manage User Knobs**.
2. In the dialog that opens, select the controls that you want to delete from the list and click **Delete**.
3. To delete an entire tab, select all controls on the tab as well as the tab name and click **Delete**.

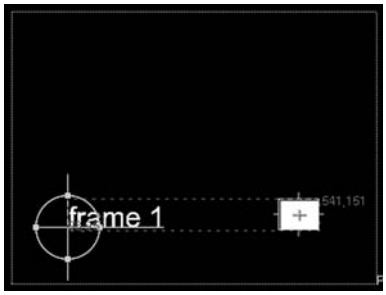
Examples

Below are some examples on how to create new controls for gizmos. To try them out, do the following preparations:

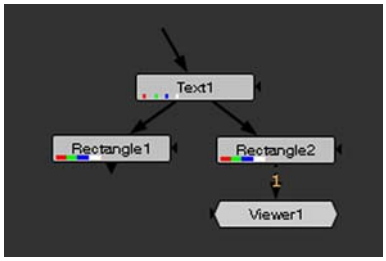
1. Select **Draw** > **Text** and **Draw** > **Rectangle**. Create the following setup:



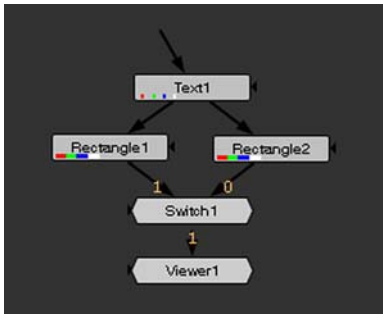
2. Double-click on the Rectangle1 node.
3. In the Viewer, resize and reposition the rectangle until it looks like the following:



4. In the Rectangle1 properties panel, go to the **Color** tab. Click on the **4** button to display multiple values rather than the slider. Enter **1** as the value for **r**, and **0** as the value for **b**, **g** and **a**. This changes the colour of the rectangle from white to red.
5. Copy the Rectangle1 node and paste it into the same script. Create the following connections:



6. Double-click on the Rectangle2 node and change the colour of the rectangle from red to green (r 0, g 1, b 0, a 0).
7. Select **Merge** > **Switch** to add a Switch node. Create the following connections:



8. Select the Text1, Rectangle1, Rectangle2 and Switch1 nodes and press **Ctrl/Cmd+G** to group them. This group will be the gizmo we will add controls to in the following examples.
9. Delete the original four nodes from the **Node Graph** tab.
10. Select the Group node and append a Viewer to it.

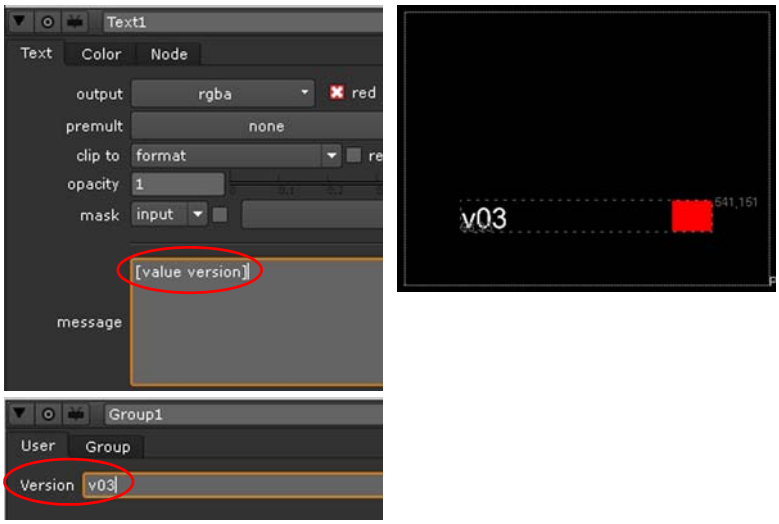
Example 1

In this example, we add a control called **Version** to the Group node controls. This control is an input field. Whatever is entered in the field is called by the Text1 node and displayed in the Viewer when you view the output of the group.

1. Open the Group properties panel and right-click on the dark grey background. Select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Text input Knob** to add a text input field control to your Group properties panel.
3. Enter *version* as the **Name** for the control, *Version* as the **Label**, and *Enter the version number here* as the **Tooltip**. Click **OK** and **Done** to close the dialogs.

This step created a tab called **User** in the Group node controls. All the controls you add or pick are added on this tab by default. As you can see, the **Version** control is now there.

4. On the **Group1 Node Graph** tab, double-click the Text1 node to open its controls. In the **message** field, enter the following expression: **[value version]**. This expression calls the control named *version* that you created in the previous step. Therefore, whatever is entered in the **Version** field of the Group node (for example, **v03**), will appear as a result of the Text1 node.



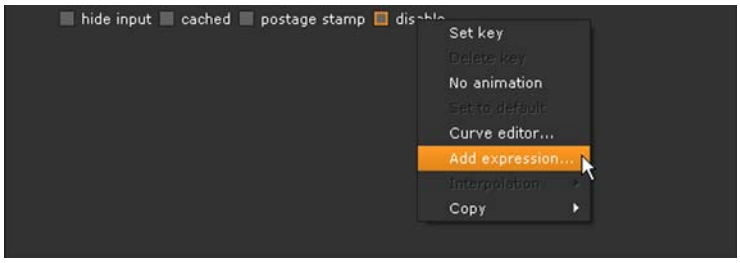
Example 2

This example teaches you to create a checkbox control that the users can use to specify whether they want to display or hide the version number added in the previous example.

1. In the Group properties panel, right-click on the dark grey background and select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Check Box** to add a checkbox control to your Group properties panel.
3. Enter *hideversion* as the **Name** for the control, *Hide version number* as the **Label**, and *Check this to hide the version number* as the **Tooltip**.
4. To have the new control appear next to the Version control (created in the previous example) rather than below it on its own line, uncheck **Start new line**. Click **OK** and **Done** to close the dialogs.

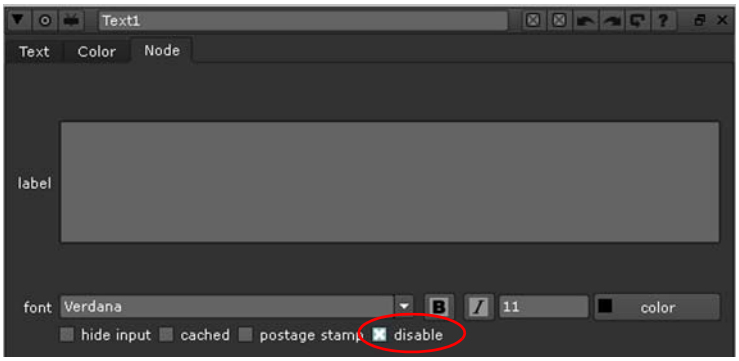
   The control you created appears in the Group properties panel now.

5. In the Text1 controls, go to the **Node** tab. Right-click on the **disable** control and select **Add expression**.



6. In the **Expression** field, enter **hideversion** (or, if you want to make it clear that the control is in the enclosing group, you can also use **parent.hideversion**). This calls the control you created in steps 2 and 3. Click **OK**.

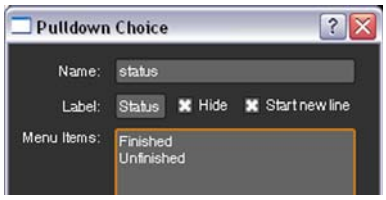
From now on, whenever **Hide version number** is checked in the Group controls, the Text1 node is disabled and you cannot see the version number it would otherwise create.



Example 3

In this example, we add we add a control labeled **Status** to the Group controls. This control is a pulldown menu with two options: **Finished** and **Unfinished**. When **Finished** is selected, the green rectangle is displayed. When **Unfinished** is chosen, you'll see the red rectangle instead.

1. In the Group properties panel, right-click on the dark grey background and select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Pulldown Choice** to add a pulldown menu control to your Group properties panel.
3. Enter *status* as the **Name** for the control and *Status* as the **Label**. In the **Menu Items** field, list the items you want to appear in the pulldown menu - in this case, *Finished* and *Unfinished*.



Finally, enter *Select the production status here* as the **Tooltip**. Click **OK** and **Done** to close the dialogs. The **Status** control should now appear in the Group controls.

4. On the **Group1 Node Graph** tab, double-click the Switch1 node to open its controls. Right-click on the **which** field and select **Add expression**.
5. In the dialog that opens, enter the following expression: **status==0** (or, **parent.status==0**). This expression calls the control named *status* that you created earlier in this example. For the pulldown menus, the first item is **0**, the next **1**, the next **2**, and so on.

   From now on, whenever **Finished** is selected under **Status**, the green rectangle is shown. When **Unfinished** is chosen, the red rectangle is shown.

Example 4

This example teaches you how to visually group and rearrange the controls you created for the Group properties panel. You can do this by renaming the **User** tab, and using static text and divider lines to group the controls on the tab.

First, we'll rename the **User** tab in the Group properties panel:

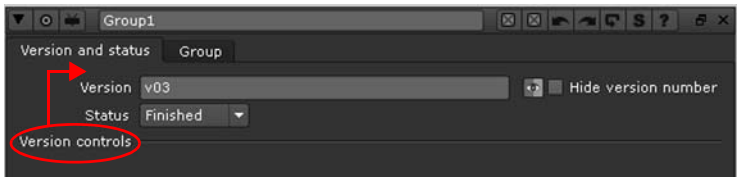
1. In the Group properties panel, right-click on the dark grey background and select **Manage User Knobs**.
2. In the dialog that opens, select **User** and click **Edit**.
3. In the **Label** field, enter a new name for the tab, for example, *Version and status*. Click **OK** and **Done** to close the dialogs.

   If you now look at the Group controls, you'll notice that the **User** tab has been renamed to **Version and status**.

Next, we'll group the two version controls of the Group node under a title called *Version controls*:

1. In the Group properties panel, right-click on the dark grey background and select **Manage User Knobs**.
2. In the dialog that opens, select **Add** > **Text** to add text to your Group properties panel.
3. Enter *versioncont* as the **Name** for the control and *Version controls* as the **Label**. Click **OK** and **Done** to close the dialogs.

   This adds the text *Version controls* to the Group properties panel. However, the text does not appear where we want it to appear: on top of the **Version** and **Hide version number** controls. Let's move it up.



4. Right-click on the Group properties panel again and select **Manage User Knobs**.
5. Select **[Version controls]** from the list and click **Up** three times. Click **Done**.

   The text should now appear on top of the Group properties panel, above the version controls.

Finally, we'll add a divider line between the version controls and the **Status** control:

1. In the Group properties panel, right-click on the dark grey background and select **Manage User Knobs** again.
2. In the dialog that opens, select **Add** > **Divider Line** to add a line to divide the controls in your Group properties panel.
3. Select the line from the Manage User Knobs dialog, where it is shown as **unnamed**.
4. Click the **Up** button once to move the line between the **Hide version number** and **Status** controls. Click **Done**.

   If you now open the Group controls, you'll notice that a there's a line between these controls.

Hopefully, the above examples have given you an idea of how to create user controls for your gizmos. Once you have all the controls you need, remember to save your gizmo (for instructions, see step 5 under *Creating Gizmos* on page 420).

Sourcing Gizmos

To source a gizmo:

Create a menu option referencing the gizmo (see *Defining Custom Menus and Toolbars* on page 414).

OR

Instruct artists to invoke the gizmo by:

- typing **x** on the Node Graph or Properties Bin and entering the gizmo name (without the extension) as a TCL command in the dialog that opens.
- opening the Script Editor and entering **nuke.load ("*gizmo name*")** where ***gizmo name*** stands for the name of the gizmo without the extension.
- selecting **Other** > **All plugins** > **Update** and once this is done pressing **Tab** on the Node Graph and entering the gizmo name.

Custom Plug-ins

The Nuke developer's kit (NDK) allows developers to create and compile their own binary plug-ins.

To source a custom plug-in:

1. Place the plug-in file in the plug-in path directory. Its name should include a **`.dll`** (on Windows), **`.so`** (on Linux) or **`.dylib`** (on Mac) extension.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Create a menu option referencing the plug-in file (see *Defining Custom Menus and Toolbars* on page 414).

   Or instruct artists to invoke the plug-in by opening the Script Editor and entering **nuke.load ("plug-in name")** where **plug-in name** stands for the name of the plug-in without the extension.

Sourcing TCL Procedure

A Nuke script or gizmo is in fact a TCL procedure (script). Thus, Nuke also allows you to hand code generic TCL procedure to automate Nuke in various ways.

To source a generic TCL procedure:

1. Place the TCL procedure file in the plug-in path directory. Its name should include a **`.tcl`** extension.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

2. Create a menu option referencing the plug-in file (see *Defining Custom Menus and Toolbars* on page 414).

   Or instruct artists to invoke the TCL script by opening the Script Editor and entering **nuke.load ("procedure/script file name")** where **procedure/script file name** stands for the name of the procedure of script file without the extension.

Tip

For some code samples of useful Nuke TCL procedures, look inside the **[Nuke directory]/plugins** directory.

Template Scripts

You can create a template script that is loaded instead of an empty script every time you launch Nuke or select **File** > **New** or **File** > **Close**. This allows you to save lookup table (LUT) setups and favourite arrangements of nodes, for example.

To create and use a template script:

1. Create the script you want to use as a template.
2. Select **File** > **Save as**. Navigate to **~/.nuke**. The tilde (~) stands for your home directory and the full stop (.) for a hidden folder.
3. Name your script **template.nk** and click **Save**.

The next time you launch Nuke or select **File** > **New** or **File** > **Close**, Nuke loads the template from **~/.nuke/template.nk**.

Tip

If you're not sure of the location of your home directory, on Linux and Mac you can open a terminal window and type **echo $HOME**. The terminal returns the pathname to your home directory.

On Windows, you can find the .nuke directory under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable. To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you will get an error.

Here are examples of what the pathname may be on different platforms:

Linux: /users/*login name*

Mac: /Users/*login name*

Windows: *drive letter*:\Documents and Settings*login name* (XP) or drive letter:\Users*login name* (Vista)

Defining Common Preferences

The Nuke Preferences dialog (**Edit** > **Preferences**) allows any user to make myriad behaviour and display adjustments to the interface. However, you may wish assign certain default preferences for artists.

To define default preferences:

1. Select **Edit** > **Preferences** to display the Preferences dialog.
2. Modify the controls within the dialog as necessary. For descriptions of what the controls do, see *The Available Preference Settings* on page 368.

3. Click **Save Prefs**. Nuke writes the modified preferences to a file called **preferences5.nk**, which is stored inside your [home directory]/**.nuke** directory.

Tip

If you're not sure of the location of your home directory, on Linux and Mac you can go to a terminal window and type **echo $HOME**. The terminal will return the pathname to your home directory.

On Windows, you can find the .nuke directory under the directory pointed to by the HOME environment variable. If this variable is not set (which is common), the .nuke directory will be under the folder specified by the USERPROFILE environment variable. To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you will get an error.

Here are examples of what the pathname may be on different platforms:

Linux: /users/*login name*

Mac: /Users/*login name*

Windows: *drive letter*:\Documents and Settings*login name* (XP) or *drive letter*:\Users*login name* (Vista)

4. Move the resulting **preferences5.nk** file into your Nuke plug-in path directory.

   For more information on plug-in path directories, see *Loading Gizmos, Plug-ins, and TCL scripts* on page 410.

5. If you haven't already done so, create a file called **menu.py** in your plug-in path directory.
6. Add the following entry in the **menu.py** file:
   ```
   nuke.load ("preferences5.nk")
   ```

Your preferences will now act as the defaults for your artists. However, should they make changes via Preferences dialog, these changes will override your defaults.

To delete (and reset) the preferences:

1. Open a terminal (or shell) as described for your operating system at the beginning of this chapter.
2. Using the prompt, go to the .nuke directory, under your home directory.
3. Enter **pwd** to display and verify the path.

   You should see something similar to

   - /users/*login name*/.nuke (on Linux),
   - /Users/*login name*/.nuke (on Mac) or
   - *drive letter*:\Documents and Settings*login name*\.nuke (on Windows XP) or

     drive letter:\Users*login name*\.nuke (on Windows Vista).

     This is not always the case, however, because on Windows the .nuke folder can be found under the directory pointed to by the HOME environment variable or (if HOME is not set) the USERPROFILE environment variable.

To find out if the HOME and USERPROFILE environment variables are set and where they are pointing at, enter %HOME% or %USERPROFILE% into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you will get an error.

4. Enter `rm preferences5.nk` to delete the preference file.
5. Close the terminal or shell.

   The next time you launch Nuke, it will rebuild the file with the default preferences.

Altering a Script's Lookup Tables (LUTs)

Overview

A script's lookup tables are curves that control the conversion between file or device colour spaces and Nuke's internal colour space. In the curve editor, the x axis represents the input pixel values and the y axis the output pixel values (normalized to the 0-1 range). When applying LUTs, Nuke looks up the input value along the x axis to determine what the y value is to output.

Nuke provides the following set of default LUTs: **linear**, **sRGB**, **rec709**, and **Cineon**[1]. You can also create an unlimited number of additional LUTs and edit or remove existing LUTs in the script's settings.

By default, Nuke uses certain LUTs for certain file types or devices. In most cases, you do not need to touch these defaults. However, there may occasionally be cases when changing the defaults is necessary: for example, if your material has been shot with a camera that records in a custom colour space, such as Panalog. In those cases, you can change the defaults in the script's settings so that you don't need to change the colour space on each Read or Write node.

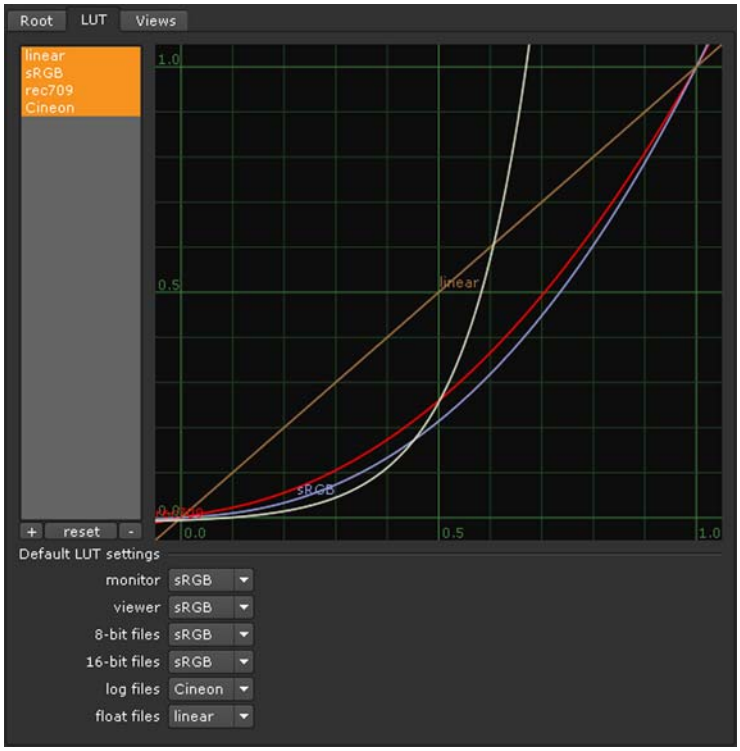
If you do not want to use the default LUT for reading or writing certain individual images, you can select the LUT to use in the corresponding Read or Write node's controls. You can also select the LUT to use for a Viewer in the Viewer's controls or settings.

Displaying, Adding, Editing, and Deleting LUTs

To display LUT curves:

1. Select **Edit** > **Project settings** to open the settings for the script.
2. Go to the **LUT** tab.
3. From the list on the left, select the LUT you want to display in the curve editor. To select several LUTs, press **Ctrl** (Mac users press **Cmd**) while selecting the LUTs. All the selected LUTs are shown in the curve editor at the same time.

1. The Cineon conversion is implemented as defined in Kodak's Cineon documentation.



To create a new LUT:

1. Select **Edit** > **Project settings** to open the settings for the script.
2. Go to the **LUT** tab.
3. Click the plus button (**+**). A dialog opens.
4. Enter a name for the new LUT and click **OK**.
5. Adjust the lookup curve to suit your needs. Click on the curve to select it. **Ctrl/Cmd+Alt+click** to add points on the curve, and drag the points to a new position. To change the shape of the curve, adjust the tangent handles.

The new LUT is now available in the global LUT settings, the **colorspace** pulldown menu of Read and Write nodes' properties panels, the Viewer controls, and the **display_lut** pulldown menu of Viewer settings.

To edit LUTs:

1. Select **Edit** > **Project settings** to open the settings for the script.
2. Go to the **LUT** tab.
3. From the list on the left, select the LUT you want to edit.
4. Adjust the lookup curve to suit your needs. Click on the curve to select it. **Ctrl/Cmd+Alt+click** to add points on the curve, and drag the points to a new position. To change the shape of the curve, adjust the tangent handles.

To use the usual editing commands, such as copy and paste, right-click on the curve editor and select **Edit**. Then, select the editing command you want to use, just like you would on any curve editor.

Note

Renaming existing LUTs is currently not possible. If you want to rename a LUT, you need to add and name a new LUT, copy the information from the old LUT into the new one, and then remove the old LUT.

To reset the LUT curves back to their initial default shapes:

1. Select **Edit** > **Project settings** to open the settings for the script.
2. Go to the **LUT** tab.
3. From the list on the left, select the LUT you want to reset. To select several LUTs, press **Ctrl** (**Cmd** on a Mac) while selecting the LUTs.
4. Click **reset**.

To remove LUTs:

1. Select **Edit** > **Project settings** to open the settings for the script.
2. Go to the **LUT** tab.
3. From the list on the left, select the LUT you want to remove. Only remove LUTs that you have, for example, created by accident and are not using in your script. To remove the LUT, click the minus button (**-**).

The LUT is removed from the LUT settings, the **colorspace** pulldown menu of Read and Write nodes' properties panels, the Viewer controls, and the **display_lut** pulldown menu of Viewer settings.

Note

If you remove a LUT that is used in a node, the node continues to refer to the LUT by name and raises an error.

Selecting the LUT to Use

To select the LUT to use when reading or writing an image:

1. Double-click to open the Read or Write node's properties panel.
2. From the **colorspace** pulldown menu, select the LUT you want to use. To use the default LUT defined in Nuke's settings for the image type in question (see *Default LUT settings* below), select **default**.

To select the LUT to use for a Viewer:

Select the LUT from the Viewer's controls.
OR

1. Right-click on the Viewer and select **Viewer Settings** to open the Viewer's settings.

In the **display_lut** pulldown menu, select the LUT you want to use.

Default LUT settings

By default, Nuke uses the following LUTs in the following cases:

File Type / Device	Default LUT
monitor. This is used for postage stamps, OpenGL textures, the colour chooser display, and all other non-Viewer image displays. Also used by Truelight nodes when conversion to or from monitor colour space is required.	sRGB
viewer. This is used for Viewers. If you are using a Viewer Input process that converts all the way to monitor-correct space, you can set this to **linear** to effectively disable the additional change, but leave **monitor** to **sRGB** to get reasonably monitor-correct output on non-Viewer images.	sRGB
8-bit-files. This is used when reading or writing image files that contain 8-bit data. Also used by the Merge node's sRGB switch, and to convert Primatte and Truelight nodes' inputs into sRGB and their outputs from sRGB. The Cineon LUT is also used to control the input to Truelight.	sRGB
16-bit files. This is used when reading or writing image files that contain 16-bit integer data (not half float).	sRGB
log files. This is used when reading or writing .cin or .dpx files. Also used by Truelight nodes when conversion to or from log colour space is required.	Cineon
float files. This is used when reading or writing image files that contain floating-point data.	linear

To change the Default LUT settings:

1. Select **Edit** > **Project settings** to open the settings for the script.
2. Go to the **LUT** tab.
3. From the pulldown menus under **Default LUT settings**, select the LUTs you want to use by default for each file type or device.

The new defaults are now used for any LUT setting where you have not selected a specific LUT. Any controls you have set to a specific LUT (that is, not set to **default**) will continue to use the selected LUT, and only those set to **default** will be affected.

Example Cases

Below are some examples of situations where you might need to alter the default LUTs.

Working in video colour space

Emulating compositor software that works in video colour space is not recommended. However, if you do need to do so, do the following:

1. Select **Edit** > **Project settings** and go to the **LUT** tab.

2. Under **Default LUT settings**, change the **monitor**, **viewer**, **8-bit files**, and **16-bit files** values to **linear**.

This prevents Nuke from converting from sRGB into linear. Nuke's nodes still assume linear data, but the image processing is applied to your unlinearised video colour space images.

Linear Data in 16-Bit Files

Some facilities use linear data in 16-bit files. If this is the case in your facility, do the following:

1. Select **Edit** > **Project settings** and go to the **LUT** tab.
2. Under **Default LUT settings**, change the **16-bit files** value to **linear**.

Cineon Displays

Some facilities have adjusted their monitor electronics to correctly display Cineon data. If this is the case in your facility, do the following:

1. Select **Edit** > **Project settings** and go to the **LUT** tab.
2. Under **Default LUT settings**, change the **monitor** and **viewer** values to **Cineon**.

Colour Management

Although true colour management requires using the Truelight or other nodes as VIEWER_INPUT nodes, it may be useful to approximate it with a LUT that is used for the **monitor** setting. This way, texture maps and postage stamps resemble the final display more accurately.

If your colour management is creating a monitor-corrected image, you'll want to set **viewer** to **linear** so that it has no change, and **monitor** to **sRGB** so you get reasonably monitor-correct output on non-Viewer images.

TUTORIALS

Welcome to the Tutorials! If you've reviewed the Reference chapters— which we highly recommend—you already know something about Nuke. These tutorials show how to pull everything together through a series of practical examples. We will add further tutorials as time goes on. They will appear on our web site between releases of Nuke.

The Projects

- **Tutorial 1: Compositing Basics**. Explains the Nuke user interface, project workflow, and basic compositing tasks.
- **Tutorial 2: Tracking, Stabilising and Matchmoving**. Demonstrates how to track image patterns, stabilise footage, lock down images for cleanplates and matchmove.
- **Tutorial 3: Keying and Mattes**. Shows how to pull mattes with standard keying tools and Nuke's own image-based keyer.
- **Tutorial 4: 3D Integration**. Shows how you can use Nuke's 3D workspace to help your 2D compositing.

Installing the Project Files

Before you continue, download the tutorial project files from The Foundry website and move them to a directory you'll create, called "Nuke_Tutorials". It's up to you where you put your tutorial files, but here's our recommendations below depending on your operating system.

Tip

If you're using a Mac or Linux system, log in under administrator privileges to avoid issues with permissions when installing the files.

Whatever you do, you'll need to remember where you put these files.

To create the tutorial directory (Windows):

1. On the Windows desktop, double-click the **My Computer** icon to open a file browser.
2. Double-click on **Local Disk (C:)** and open the **c:\Documents and Settings\All Users\Application Data** folder.
3. Click the right-mouse button over the displayed directory and choose **New > Folder** from the pop-up menu.
4. Name the folder: **Nuke_Tutorials**.

To create the tutorial directory (Mac OS or Linux):

1. Open a shell or terminal window.

2. At the command line, enter **mkdir ~/Nuke_Tutorials** to create the tutorial directory under your user or "home" directory.

To download and install the project files:

1. Open an internet browser and go to: *http://www.thefoundry.co.uk.*
2. Navigate to the Nuke product page from the Products menu at the top of the home page.
3. Click on the Tutorials & Example Images link on the right hand side. You can also access this page by selecting **Help** > **Tutorials** in Nuke.
4. Click the links to download the project files to your local computer.
5. Extract the downloaded files and move (or copy) them to the Nuke_Tutorials directory you created earlier.

You're now ready to start the first tutorial with Nuke.

Tutorial 1: Compositing Basics

Hello! This tutorial is your introduction to Nuke, where you'll create a simple composite and breeze through most of the windows, on-screen controls, and other user interface items.

We've heard rumours that many people would rather march through icy rain than review an introductory tutorial on digital compositing. Certainly, that's not you. When you finish this lesson you'll have a good understanding of the Nuke workflow and should feel confident about approaching the other tutorials.

Figure 1-1: Your first composite in Nuke

Before you get into the project, we have some administrative things to do—such as defining a few application preferences and project settings. We know this sort of thing is not terribly exciting, but it is terribly important, so please be patient and we'll get through it as quickly as possible.

Note

If you haven't already downloaded and moved the tutorial project files to the Nuke_Tutorials directory you created, turn to page 438 for instructions.

Starting Nuke

The Nuke icon may appear on your desktop. If so, double-click it to launch the application. Otherwise, start Nuke with one of the methods described below, assuming you have installed Nuke to the default location.

To launch Nuke under Windows:

- From the **Start** menu, choose **All Programs > The Foundry**, and then select **Nuke5.1v5**.

To launch Nuke under Mac OS X:

- Open the /**Applications/Nuke5.1v5/** folder and double-click the **Nuke5.1v5** icon.

To launch Nuke under Linux:

- Open the **/usr/local/Nuke5.1v5/** folder and double-click the **Nuke5.1** icon.

Tip

If you're operating under Linux, you can also launch Nuke from the command line of a terminal. Simply navigate to the Nuke directory and enter the name of the nuke application.

A clean copy of the main Nuke window appears. Divider lines organize the window into different panes. Each pane has one or more pages of content, separated by tabs at the top of the pane. The Toolbar appears at the left edge of the main window.

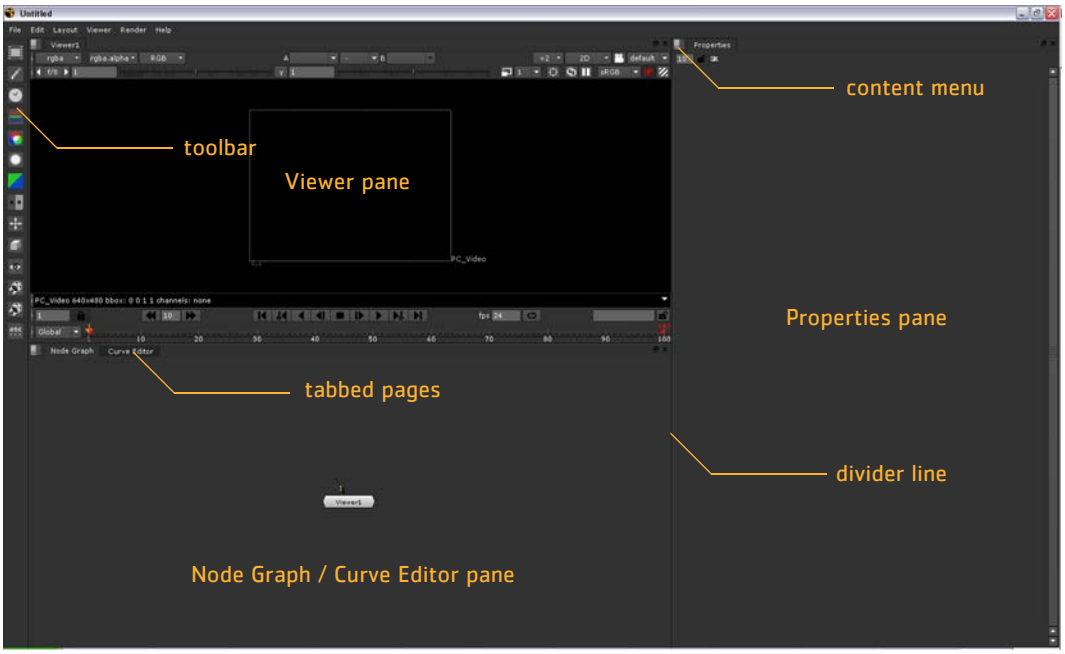


Figure 1-2: The main workspace.

By default, the panes are setup to display the Viewer, the Node Graph/Curve Editor, and Properties. You'll create the script for this project inside the Node Graph page on the Node Graph/Curve Editor pane. We'll talk about each of these onscreen controls when you need them for the project.

Using the Toolbar

The Toolbar includes the options you can use to build your project, such as importing images, layering images, drawing shapes and masks, applying colour correction, and so on. Each Toolbar icon displays a menu of *operators* or *nodes* that you can select. Roll the mouse pointer over the Toolbar and you'll see pop-up tool tips that identify each icon.

Using the Menus

The Nuke menu bar appears at the top of your screen, outside the main window. This menu begins with the options "File," "Edit," "Layout," and so on. When instructed to do so, make selections from the menu bar, or click the right mouse button to choose from a pop-up version of the menu bar.

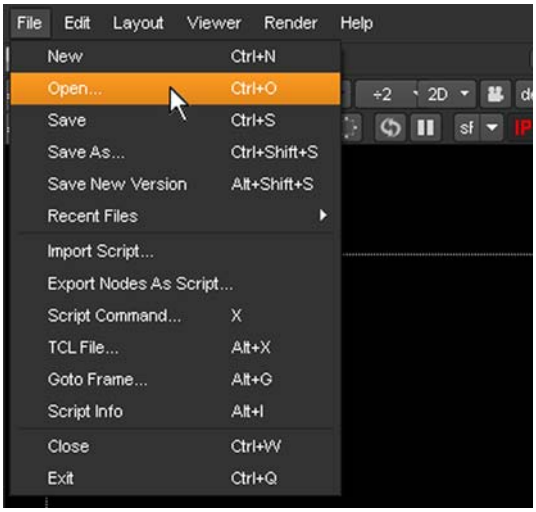


Figure 1-3: The Nuke menu bar

The "right-click" menu is highly contextual. Its options change according to the location of the mouse pointer. Right-click over the Node Graph, for example, and you'll see the options from the menu bar and the nodes you can insert from the Toolbar. Right-click over the Viewer pane and you'll see a menu of Viewer options.

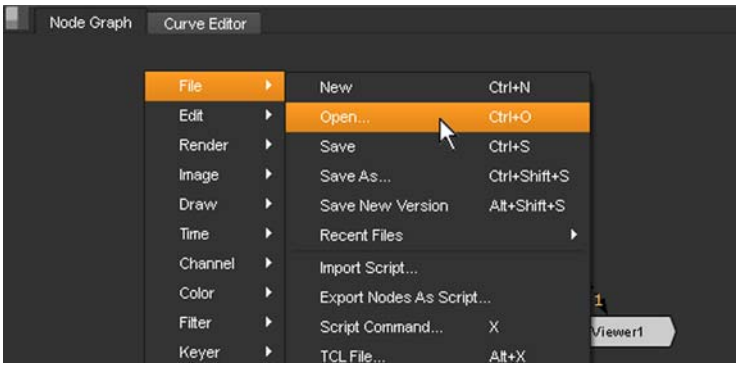


Figure 1-4: The "right-click" menu

Try the right-click menu when you can't find appropriate controls or menu options. Many features are hidden in the pop-up menu until you're in the situation where you need to use them.

Note

Nuke's menu bar, at the top of the screen, is organized a little differently between the operating systems, but the right-click menu contains the same options, regardless of the system you're using to run Nuke.

Customizing Your Layout

Nuke gives you several options for customizing the window layout. It's time for you to claim your copy of Nuke and make it your own! You don't need to customize the layout for this lesson, but why not try it now for your own personal amusement? Here are some things you can do to reorganize the window layout:

- Drag a divider line between panes to change the size of the panes.
- To divide a pane, click on the content menu (the checkered box at the upper-left corner of each pane), and choose **Split Vertical** or **Split Horizontal**.
- To add a new tabbed page to a pane, click on the content menu and choose one of content options, such as **New Viewer** or **Curve Editor**.
- Click on the "x" inside a tab to discard a tabbed page.
- To move a tabbed page, drag the tab to another pane inside the main window.
- To tear-off a page as a floating window, drag the tab outside the borders of the main window, or simply **Ctrl**+click (Mac users **Cmd**+click) on the tab name.
- Drag a floating window into a pane, inside the main window, to convert it to a tabbed page.
- From the menu bar, choose **Layout > Save Layout** *x* to save the current layout. Choose **Layout > Restore Layout** *x* to apply a previously-saved layout.
- To select a predefined colour scheme, click the right mouse button and choose **Edit > Preferences**. Then click the **Choose a Preset** button and select a colour scheme.

- Define other appearance options, such as window colours and fonts, by changing the settings under **Edit > Preferences >** the **Node Graph** tab.

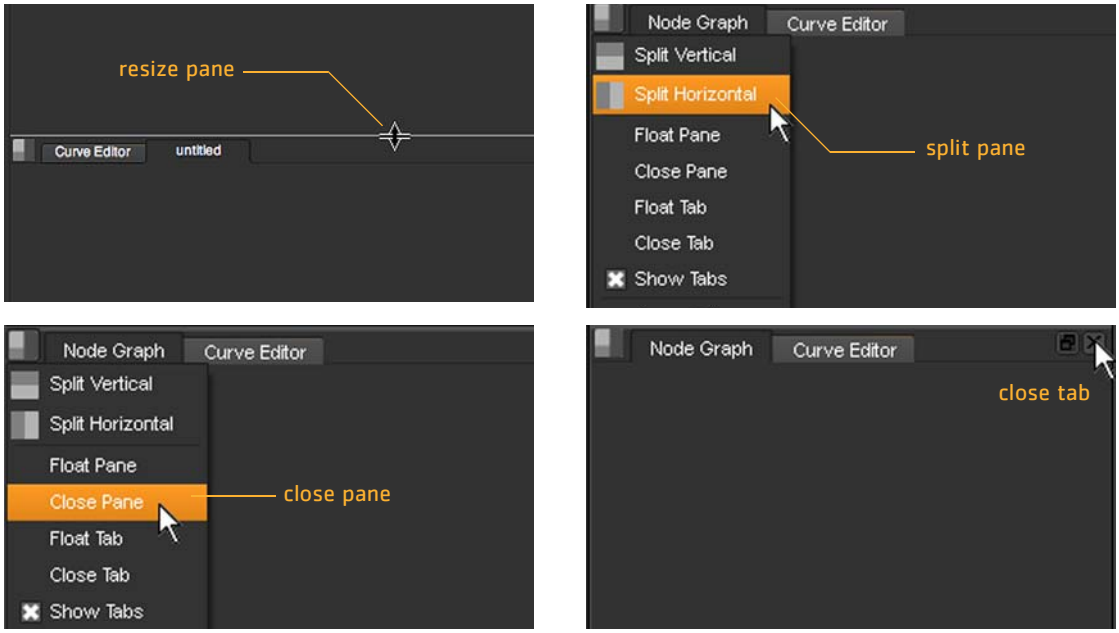


Figure 1-5: Options for customizing the window layout

Saving Files and File Backup

We assume you already know how to save files (Hint: choose **File > Save As**). In addition, Nuke includes an autosave feature, which helps recover project files after a system failure. Yes, we know that will *never* happen to you, but in the unlikely event that it does, you won't lose your work when you have autosave enabled.

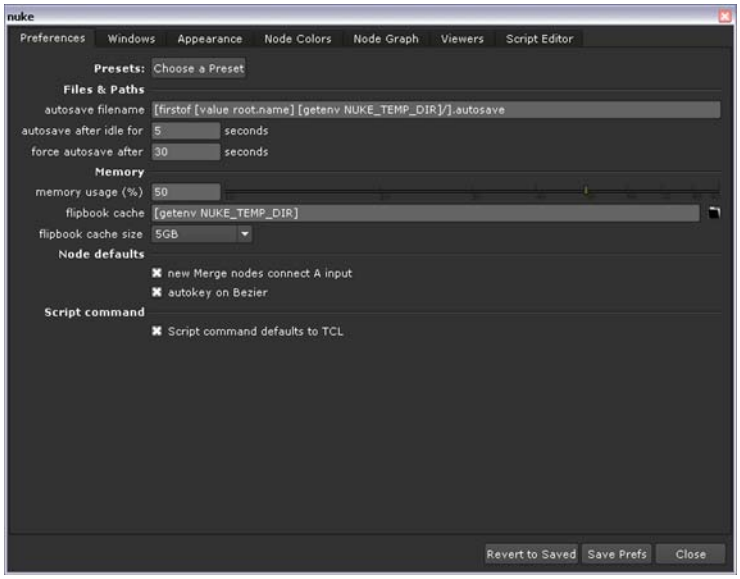
To define file/saving options:

1. Click the right mouse button over the Node Graph pane, and choose **Edit > Preferences**.

   Notice the "autosave filename" directory is set to "[value root.name].autosave." You don't need to make a change; this simply tells Nuke to store automatic backup files in the same directories as your project files.

   Now, how often would you like Nuke to generate an automatic backup while you're working? Every five minutes?

2. Change the **force autosave after** option to **300** seconds, to generate an automatic backup every five minutes.



3. Click **Save Prefs** to keep the changes and then **Close** to return to the main window.

   If you close this dialog box without clicking the Save button, then the changes will affect only the current session of Nuke.

Recovering Backup Files

You may ask, "How do I recover a backup file in the event of a system or power failure?" Good question! When you relaunch Nuke, you'll see a message that asks if you want to recover the .autosave file for the project that was last open. Click **Yes** and Nuke opens the backup file.

Tip

The .autosave files can still be useful, even when you properly exit Nuke, because they are not deleted from the directory. You can, for example, rename an .autosave file to create an archive of the previous version of your project file.

Sometimes you will see the recovery message even though you have not experienced a system failure. This happens when you exit Nuke without saving the changes to a project file, and Nuke recognizes that the time stamp on the .autosave file is later than the Nuke project file you're trying to open. In this case, you decide which version of the project file you want to open.

Turning Off Automatic Backup

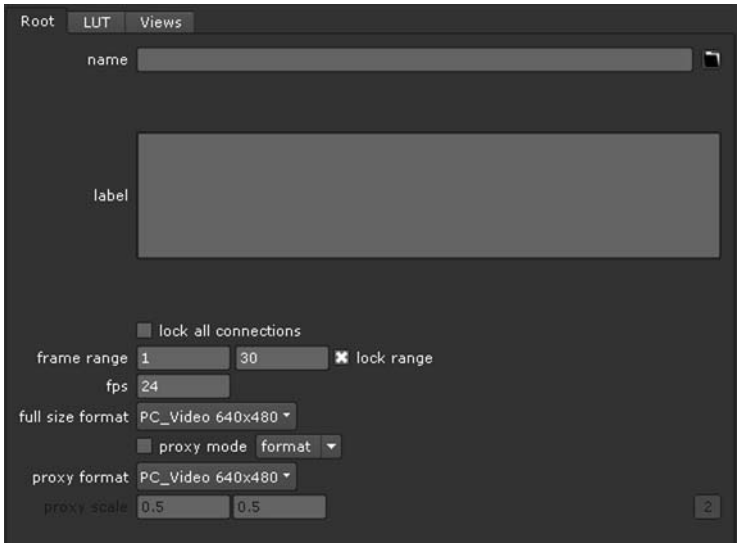
Okay. You're reading this, so we assume you're a freewheeling rebel who possibly enjoys the risk of losing your work. It's an adrenaline thing. Or perhaps you prefer to do everything yourself, manually, and you have a secret obsession for saving your files, often. Whatever the reason, you can disable the autosave features by setting the intervals for both "autosave idle" and "force autosave" to zero seconds. That's it. That's all you need to do. Good luck.

Setting Up the Project

When you start a new project, you need to define project settings for length or frame range, the number of frames per second for playback, and the output format. These options appear on the Project Settings dialog box.

To setup your project:

1. Click the right mouse button over the Node Graph, and then choose **Edit > Project Settings** from the pop-up menu.



2. In the **frame range** fields, enter a range of **1** to **28**. This will be the length of the shot we create for the project.
3. Enter **24** as the frames per second (fps).
4. Click the **full size format** list and choose **PC_Video 640 x 480**.
5. Close the **Settings** control panel.

Until now, everything you've done is standard procedure for a new project. You used the menu

Note

On the project Settings control panel, the LUT tab includes options that ensure colour integrity for your display and output devices. You don't need to change the LUT for these tutorials, but we recommend that you research and set these options for your own projects.

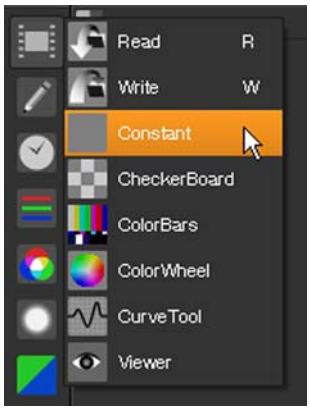
bar to access several features during the setup process, and now you'll use the Nuke toolbar to insert nodes and create a compositing tree.

Working with Nodes

A *node* is simply one of the building blocks for the list of operations you want to complete. A *node tree* is a diagram that shows the order in which the operations will be performed. Do the following to add a few nodes and start your node tree. The result will create the background for the project.

To insert nodes:

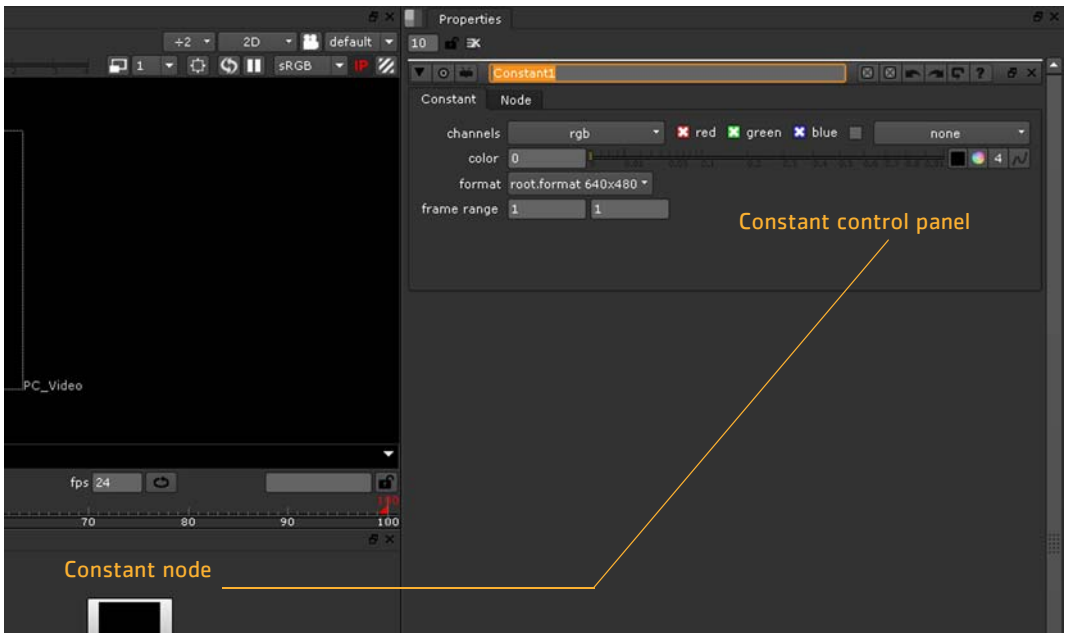
1. On the Toolbar, click the first icon to display a menu for nodes that are in the **Images** category.



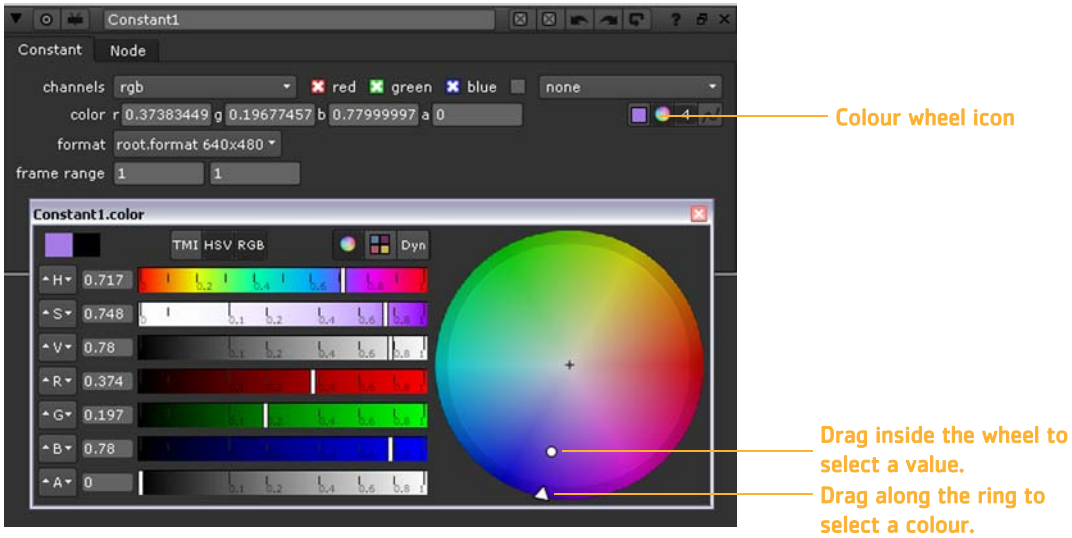
2. Select **Constant** from the menu to insert this node into the Node Graph pane.

   When you insert a new node, its control panel also appears with parameters that let you define what the node will produce. In this case, the Constant node creates a solid colour

backdrop.



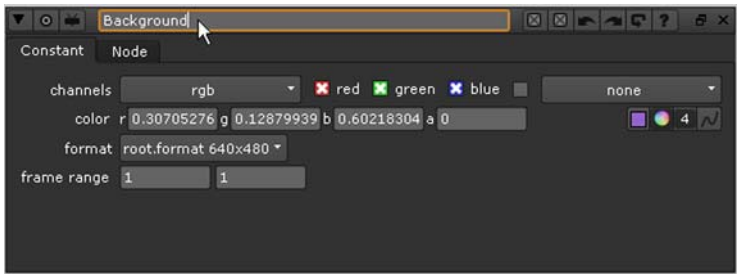
3. In the Constant control panel, click on the colour wheel to open the Colour Picker.



4. Drag the colour sliders and the cursor inside the wheel to choose a light colour, something appropriate for the "horizon" of the composite background. Then, close the colour wheel window.

   At this point, you should probably rename "Constant" to something more descriptive.

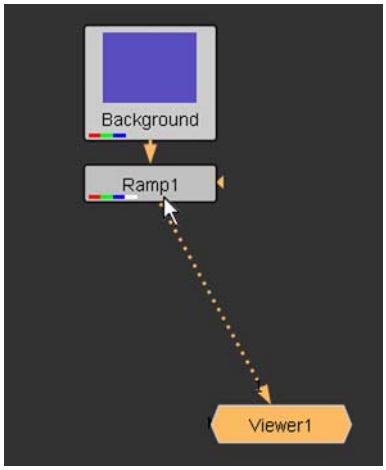
5. Inside the control panel, click on the **Constant** name. You can now edit the name, so type **Background** and press **Enter**.



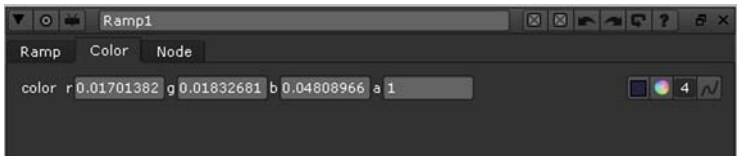
From here onward, we'll call this node the "Background" node.

6. Close the control panel for the Background node. When you need to reopen it, just double-click the node and the control panel will reappear.
7. Click on the **Background** node to select it. Then, click the right mouse button and choose **Draw > Ramp** from the pop-up menu.

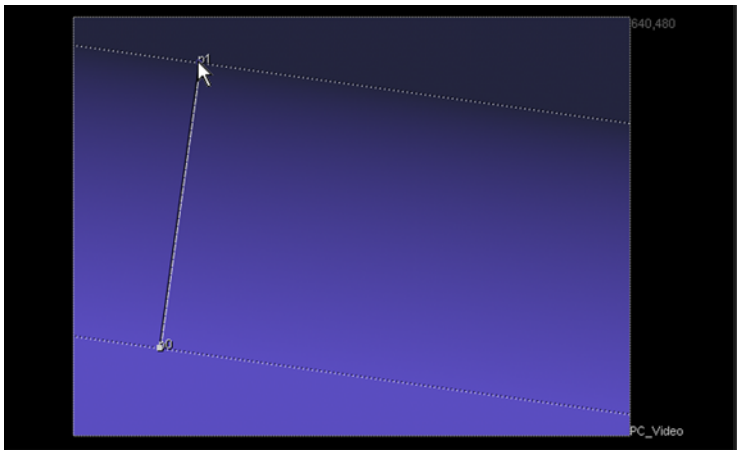
8. Drag the tail of the arrow from the **Viewer1** node to the centre of the **Ramp1** node. You'll see the output of the Background node and the ramp controls displayed in the Viewer window.



9. Click the **Colour** tab inside the control panel for **Ramp1**. Then choose a dark colour that blends well with the colour you selected for the Background node.



10. Click the Ramp tab in the control panel to reactivate the overlay controls. Then, drag the **p0** and **p1** control points to adjust the spread and angle of the ramp over the background.



11. When you're happy with the results, close the Ramp1 control panel to remove the overlay.

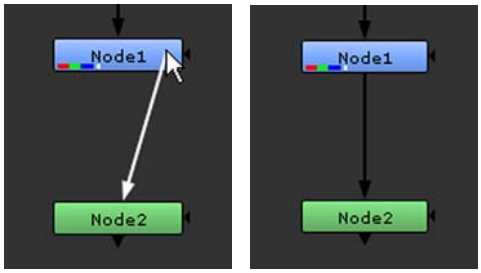
Connection Tips

Most nodes have input and output connectors that are used to establish the order in which the operations will be calculated.

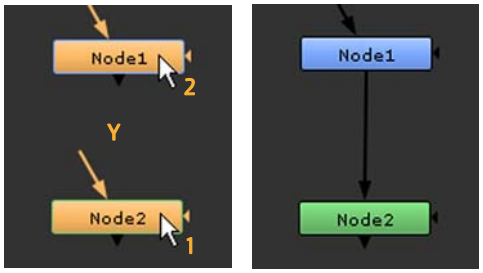
Figure 1-6: Connectors on a node.

Try the following to connect nodes after you insert them into the Node Graph:

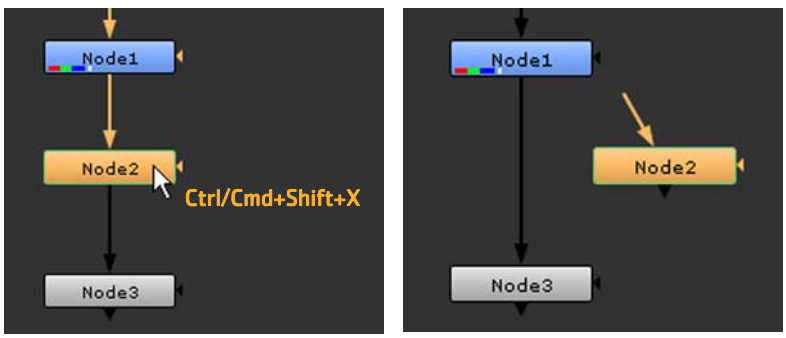
- Drag an input or an output connector onto another node to establish a connection.



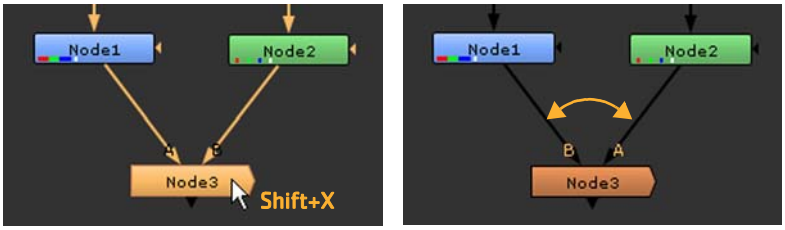
- Select a node, press the **Shift** key and select a second node. Then press **Y** or **Shift+Y** to create the connection. Try this with various nodes as the results vary according to the order in which you select the nodes and the number of input/output connectors on the nodes.
  For example, if you first select Node2 and then Node1, and press Y when both are selected, you get the following result:



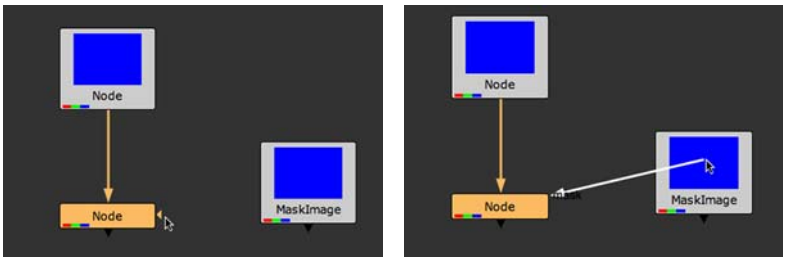
- Select a node and press **Ctrl/Cmd+Shift+X** to extract the selected node from the tree.



- For nodes that have two inputs, select the node and press **Shift+X** to swap the A/B inputs.



- Drag the mask connector to the node that provides the image you want to use as the mask for the selected node.

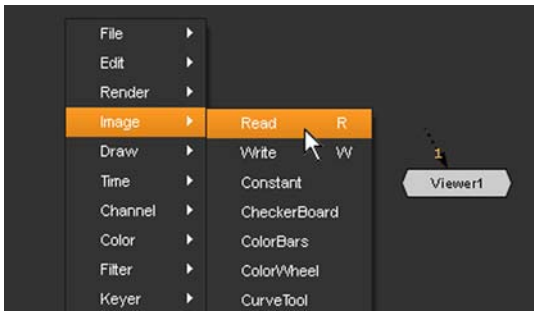


Importing Image Sequences

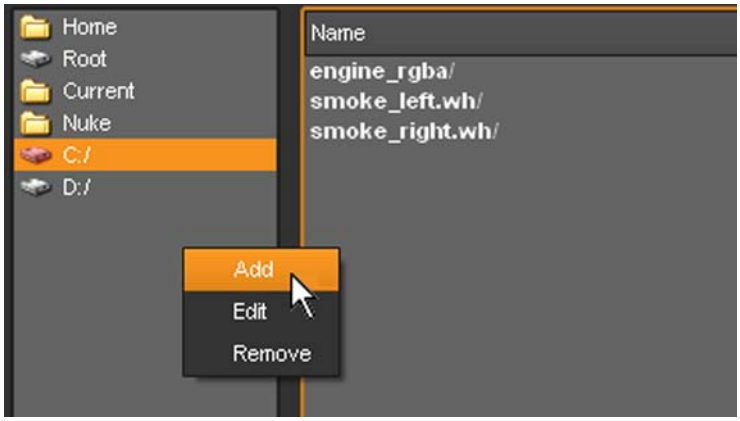
For this project, you need to import a few image sequences for the foreground elements and a background plate.

To read the images:

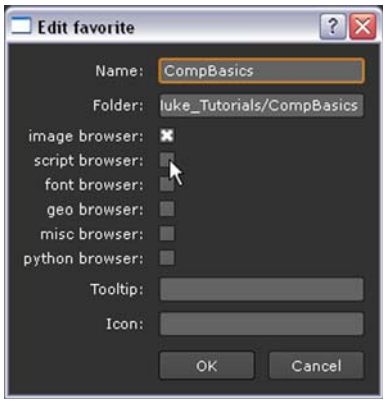
1. Click on a blank space in the Node Graph. This ensures none of the nodes are selected.
2. Click the right mouse button over the Node Graph and choose **Image > Read** from the pop-up menu.



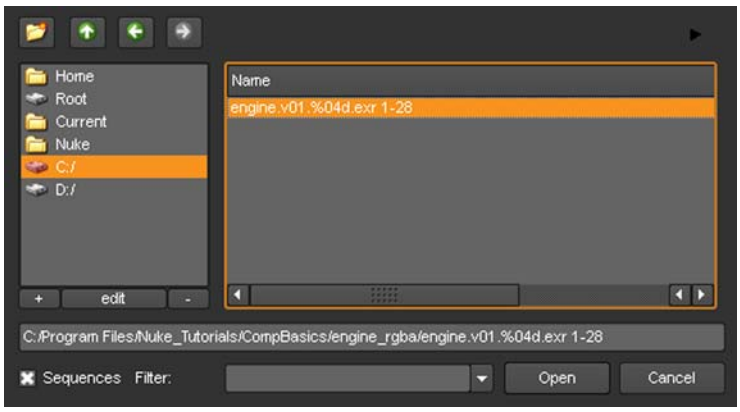
   A file browser appears. This where you select the image file you want to import. When you browse through your directories from this window, Nuke will display sequentially-numbered files as one item in the directory.
3. Browse to the **Nuke_Tutorials/CompBasics/** directory.
4. Add a bookmark to this directory. Right-click over the list, on the left side of the file browser window, and choose **Add** from the pop-up menu.



5. Type a name for the bookmark or keep the default, which is the directory name. Then click **OK**.



6. Open the **engine_rgba** directory, select the **engine.v01.%04d.exr** image sequence, and click **Open**.

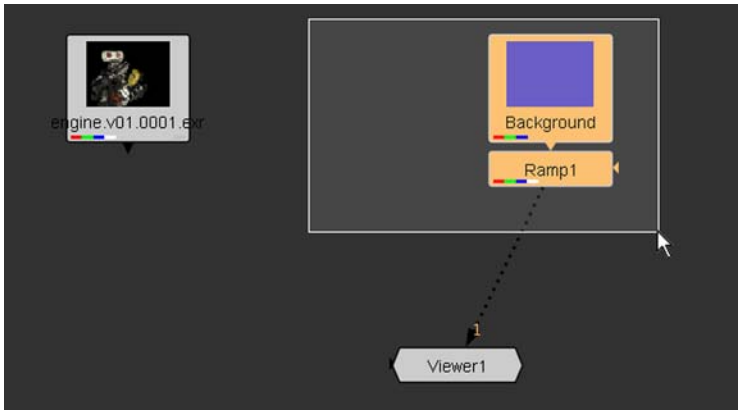


Nuke retrieves the image sequence and displays it as a thumbnail on the node. The Read control panel displays the resolution and the frame range for the image.

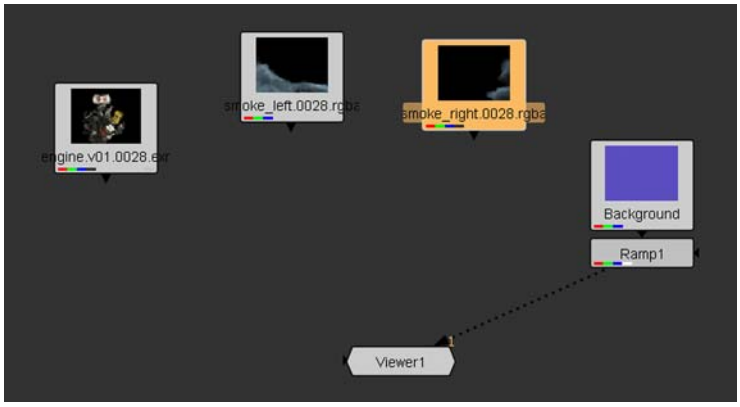
Note

Nuke reads images from their native format, but the Read node outputs the result using a linear colour space. If necessary, you can change the Colourspace option in the Read node's control panel, or insert a **Colour > Colourspace** node to select the colour scheme you want to output or calculate.

7. Drag a marquee (hold down the left mouse button while dragging) around the Background and Ramp nodes to select them. Then drag them to the right to make room for additional nodes.



8. Choose **Image > Read** from the right-click menu to import another image sequence. Use the file browser to select the image sequence stored in **Nuke_Tutorials/CompBasics/smoke_left.wh/smoke_left.%04d.rgba**.
9. Add one more Read node and retrieve the image sequence stored in **Nuke_Tutorials/CompBasics/smoke_right.wh/smoke_right.%04d.rgba**



10. Arrange the nodes, as shown above, to allow some room to create the connections for the node tree.

Navigating Inside the Windows

The node graph panel can seem very small, especially when your node tree grows. True, you already know how to resize and tear-off the windows, but sooner or later you will run out of display real estate. It's time to learn some navigation controls that will help you work in the node graph (and other windows) in Nuke. Try the following navigation controls:

Panning Your View

- Windows/Linux: While pressing the **Alt** key and the left mouse button, drag the mouse pointer across the node graph.
- Mac OS X: While pressing the **Option (alt)** key and the left mouse button, drag the mouse pointer across the node graph.

  As you drag the mouse, you pan your view of the node graph.

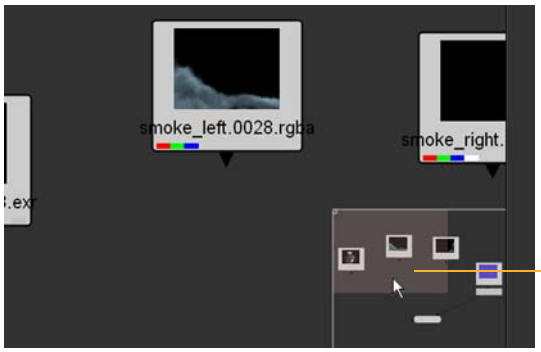
Zooming or Magnifying Your View

- Windows/Linux: While pressing **Alt** and the middle mouse button, drag the mouse pointer across the node graph.
- Mac OS X: While pressing **Option (alt)** and the middle mouse button, drag the mouse pointer across the node graph.

  Drag to the right and you'll zoom-in. Drag to the left and you'll zoom-out.
- Keyboard zoom-in/out. Tap the plus (**+**) key to zoom-in. Tap the minus key (**-**) to zoom-out.

Using the Node Graph Overview

- When the node tree extends beyond the borders of the window, a navigation box appears in the lower-right corner of the node graph. Drag the shaded rectangle inside the box and you'll quickly pan to another view of the node tree.



Framing the View in the Window

- Press the letter **F** on your keyboard to fit the entire contents of the node tree within the borders of the node graph.

The navigation controls for the node graph also work inside the next window on our agenda, the Viewer.

Working with Viewers

The *postage stamps* on the nodes—those little pictures, often called thumbnails—show what each node passes onto the next node in the tree. Although quite lovely, they won't do for real compositing work. You need to open a Viewer window to see the full picture.

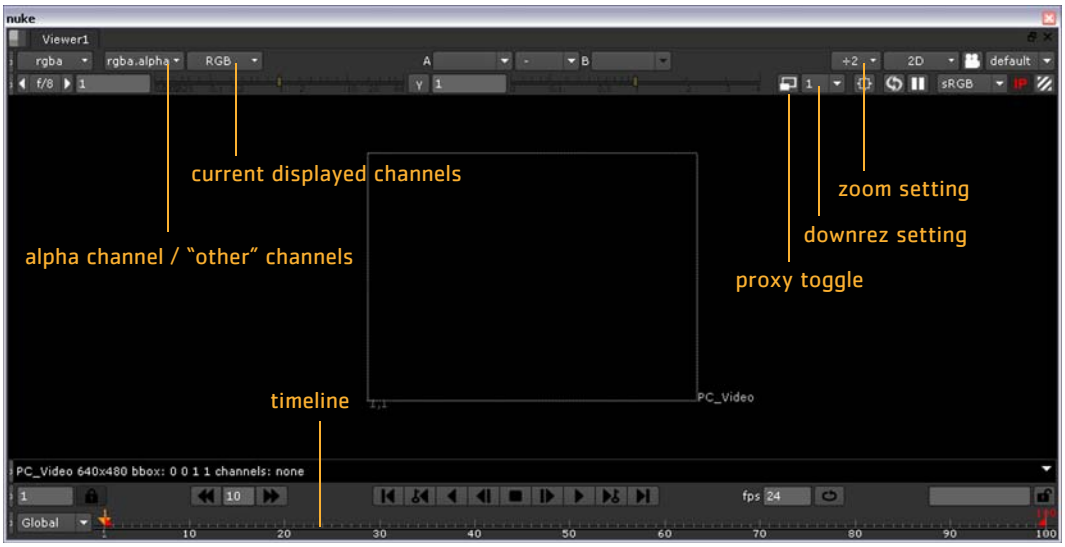


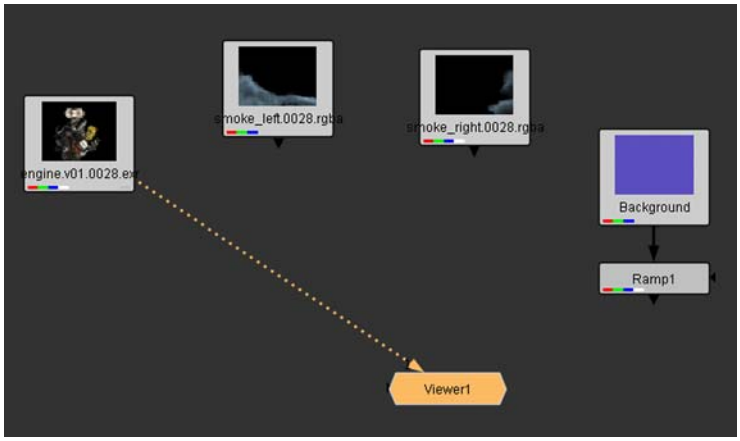
Figure 1-7: A Viewer window,

You can open several Viewers at once. In addition, you have up to 10 pages, or buffers, for each Viewer window; these allow you to toggle between different views along the node tree.

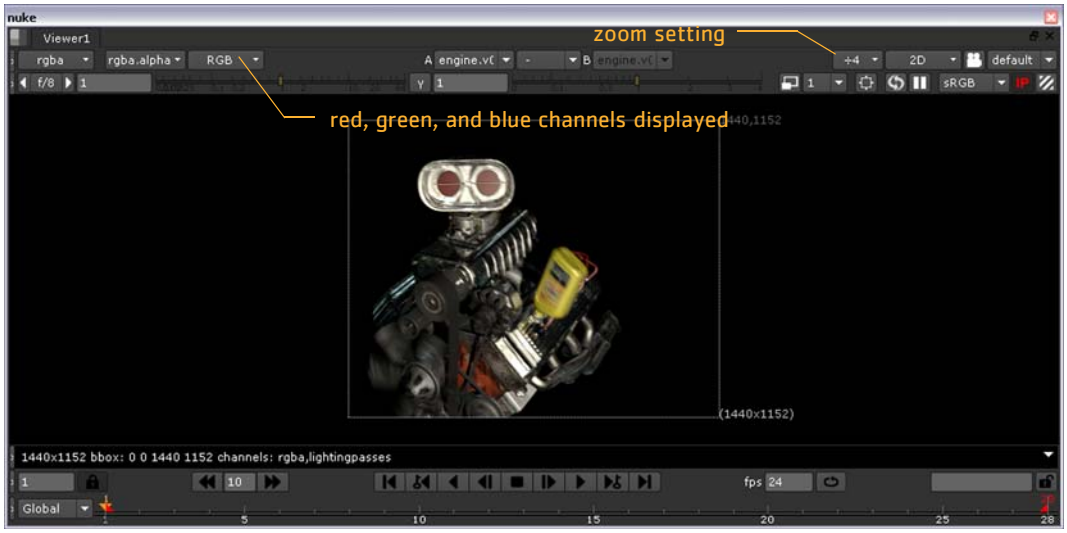
When you start Nuke, you will see a default Viewer node in the Node Graph. You can easily drag the connection arrow from a node onto the Viewer to display the node's output. You can open additional Viewers by choosing **Viewer** > **Create New Viewer** from the menu bar or by pressing **Ctrl+I**.

To display the images in a Viewer window:

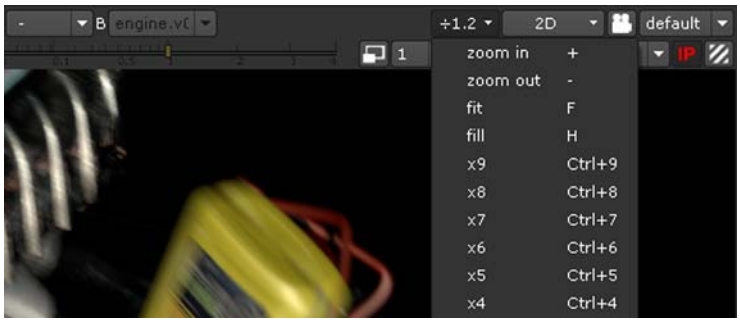
1. Drag the connector from the Viewer node onto the Read node for the **engine.v01** clip.



Here's an alternate method: Select the **engine.v01** clip node and then press **1** to connect to the Viewer node. Nuke displays the node's output in the Viewer window.


zoom setting
red, green, and blue channels displayed
(1440x1152)
1440x1152 bbox: 0 0 1440 1152 channels: rgba,lightingpasses

2. Press the **Alt** key (Mac users press **Option**) and the left mouse button, and drag the mouse pointer across the Viewer window to pan.
3. Press **Alt** (Mac users press **Option**) and the middle mouse button, and drag to zoom in/out. You can also use the "zoom" list at the top of the Viewer to magnify the view.



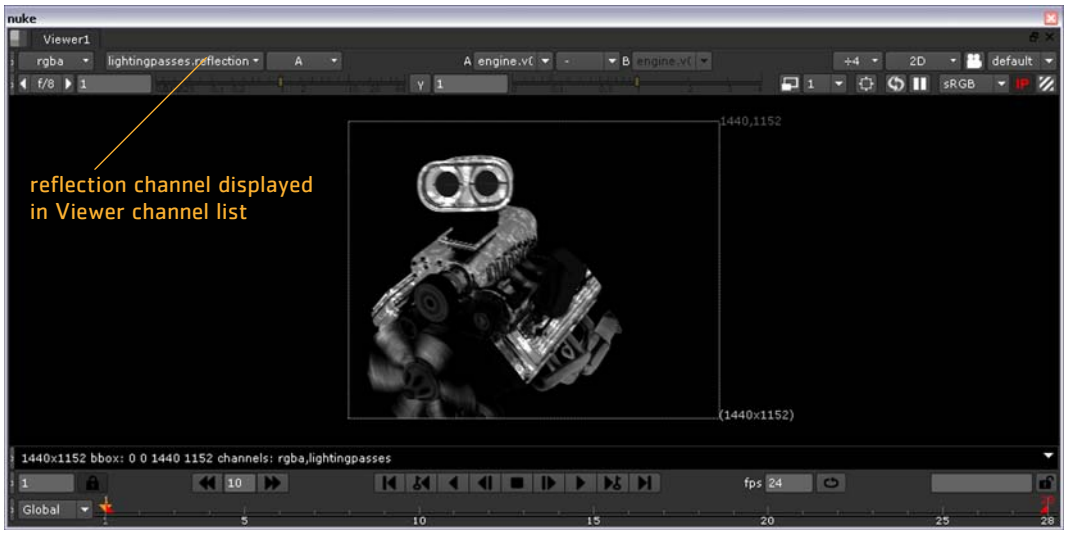
4. Press **F** to fit the current image into the borders of the Viewer window.

   This image has different *channels* of information you can view. The "RGB" label appears at the top because the Viewer now shows the result of the red, green, and blue channels.
5. To view individual colour channels, press **R** (red), **G** (green), **B** (blue) or **A** (alpha). As you press each hot key, the label at the top of the Viewer reflects the displayed channel.

6. Press one of the channel hotkeys again to return to the "RGB" display, or choose **RGB** from the Viewer's channel list.

   In addition to the standard colour channels for red, green, blue, and alpha, this image also includes channels for specular highlights, reflections, and other masks.

7. To view additional channels, press **A** to display the alpha channel, and then select the **lightingpasses.reflection** channel from the Viewer channel list.



   You now see the reflection mask from the image file.

8. Select **rgba.alpha** from the Viewer channel list to reset this as the preferred channel when you press the A key.

9. Press **A** again to toggle the display and show all colour channels.

To view multiple inputs:

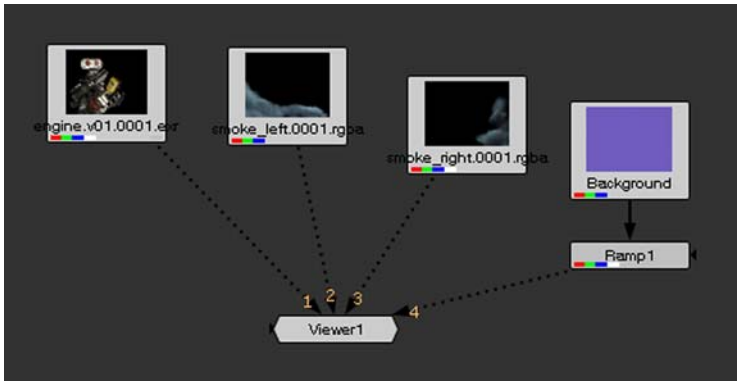
1. Select the Read node for the **smoke_left** clip, and press **2** at the top of your keyboard or on the numeric key pad.

This creates a second connection to the Viewer from the selected node. When the cursor is over the Viewer, you can press a number on the keyboard to pick the connection you want to view.

2. Move the mouse pointer over the Viewer and press **1** to display the **engine.v01** clip. Press **2** to display the result of the **smoke_left** node.

   In this manner, you can connect multiple images to the same Viewer and then switch between the images.

3. Select each of the other nodes and press a number to establish a connection to the Viewer.



4. Move the mouse pointer over the Viewer and press the numbers on your keyboard to display each of the connected nodes.

   As you switch between the different views, the images may appear to be the same size. However, if you look in the lower-right corner of the Viewer, you'll see the images have different resolutions.

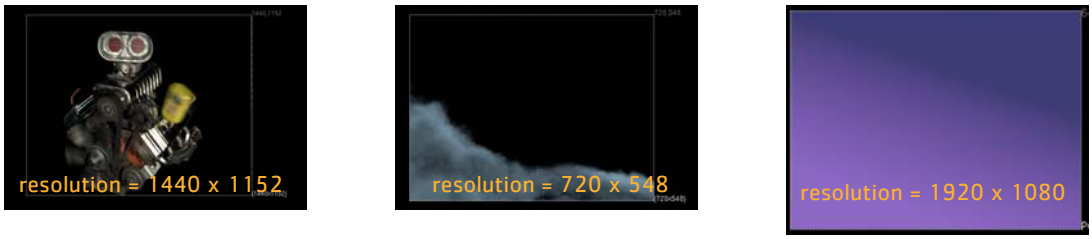


Figure 1-8: These images have different resolutions

Nuke allows multiple resolutions in one composite, but you need to conform these images to match the project resolution. This will allow the elements to be properly aligned in the composite.

Reformatting Images

Elements created within the Nuke script, such as Background and Ramp, automatically inherit the global format and that's how you want it for this project. The imported images, however, do not conform to the project settings and must be reformatted.

To conform images to the project format:

1. Click the Read node for the **engine.v01** clip to select it.

2. Click the right mouse button and choose **Transform > Reformat** from the pop-up menu.
3. Repeat steps 1 and 2 for all the Read nodes in the node graph.
4. Move the mouse pointer over the Viewer, and press the keyboard numbers (**1**, **2**, and **3**) to switch between the connected images.

   Each image should now conform to the project format.

If you change the delivery format in the project settings, then all elements set to "root.format" will also change to the new project settings. If you neglect to reformat images when you read them into the project, the images will retain their original format, independent of the project settings.

Using Proxies and "Down-res"

Proxies are low-resolution versions of the final image you intend to create. For many compositing tasks, the low-res version can help you work faster. Then, when you're ready to create the final output, switch proxy mode off and return to the full-res version.

Figure 1-9: Proxies in Nuke

Nuke can generate proxies on-the-fly, according to the *scale* or *format* of your images. You select the method under **Edit** > **Project Settings**.

To toggle the proxy resolution defined under Project Settings, you use the "proxy" button on your Viewer. Alternatively, you use the "down-res" button to lower the display resolution of individual Viewers. The down-res button works both in the full-res and proxy mode.

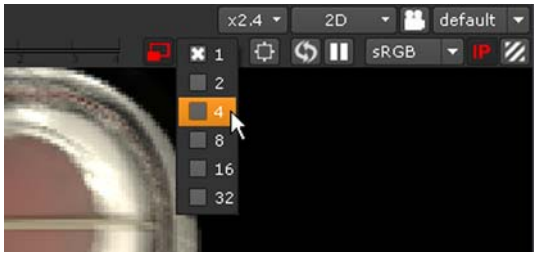
To activate proxy mode:

1. Click the right mouse button over the node graph and choose **Edit > Project Settings**.
2. Make sure the Viewer window is open.
3. Press the keystroke to toggle Proxy mode, **Ctrl+P**.

   A label inside the Viewer indicates that you are now in proxy mode.
4. Move the mouse pointer over the Viewer, and press the plus (+) key several times to zoom-in.
5. Press **Ctrl+P** a few times to toggle between hi-res and proxy mode.
6. Before you continue, press **Ctrl+P** to switch back to full resolution.

To activate "down-res":

1. Choose **4** from the "down-res" list to change the display resolution to 25% of full resolution.



With a reduced resolution, Nuke requires less time to calculate and display your images.

2. Change the "down-res" setting back to **1**, which is 100% of the active resolution.

   If you turned off the proxy mode, you should be back to full resolution. If proxy mode is turned on, the display resolution will be 100% of the proxy resolution.

Compositing Images

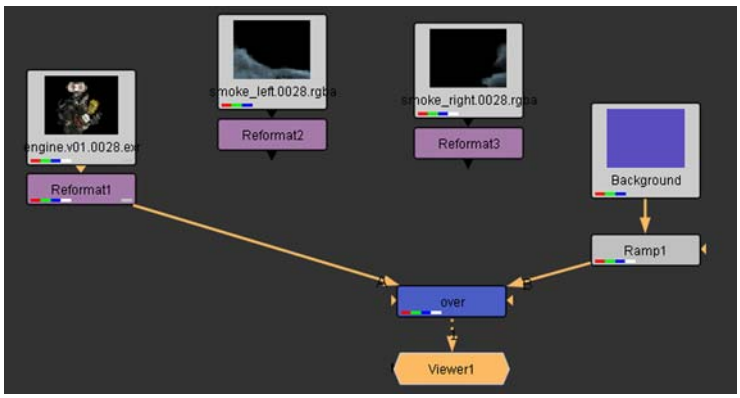
The Merge nodes create a composite with two or more images, using various compositing algorithms. In this example, we'll do a very simple "A over B" composite to layer the foreground image over the background.

You can insert a compositing node from the Toolbar or menus, but we'll show you a shortcut that bypasses both of these. The trick is the select both nodes you want to composite and then press a hot key to assign a compositing node.

To composite two nodes:

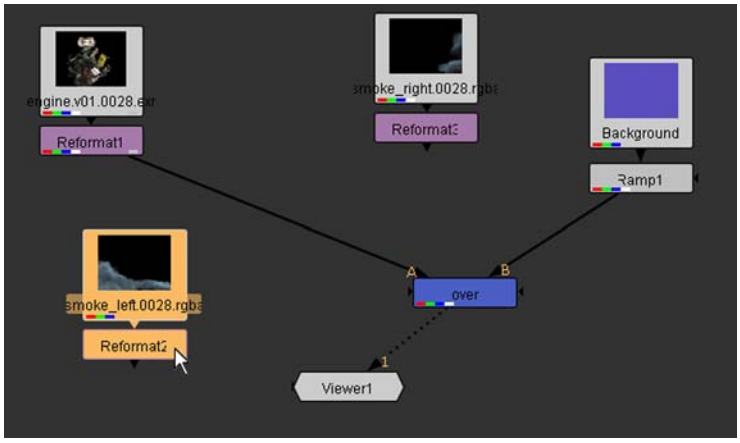
1. Select the **Reformat1** node, attached to **engine.v01**. This provides the foreground image for the first compositing operation.
2. Press the **Shift** key and select the **Ramp1** node. Both "engine.v01" and "Ramp1" nodes should be selected.
3. Press the letter **M** to insert a Merge node.

   The first node you selected is attached to the A input on the Merge node, as the foreground input. The second node you selected is attached to B, the background input. If necessary, you can swap the A and B inputs of a merge node by pressing **Shift+X**.

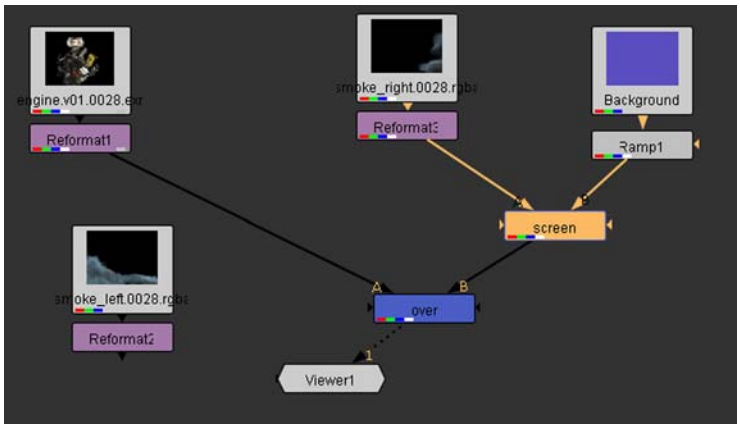


In the Merge node control panel, the **operation** parameter determines the compositing algorithm used to generate the result of the two inputs—the selected operation becomes the name of the node in the Node Graph.

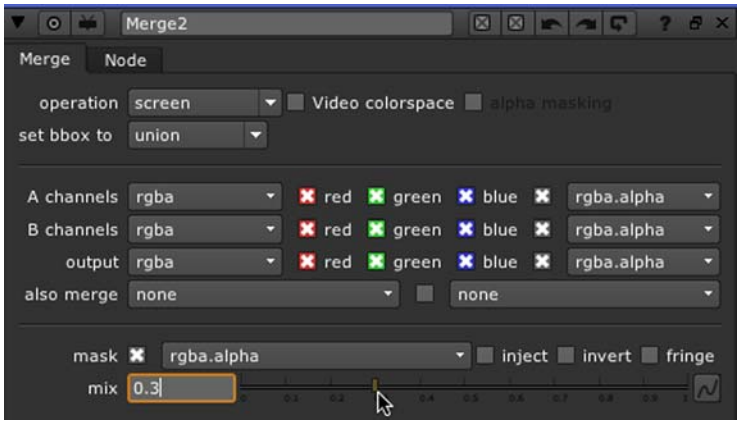
4. Rearrange the nodes, so that the node tree looks similar to this:



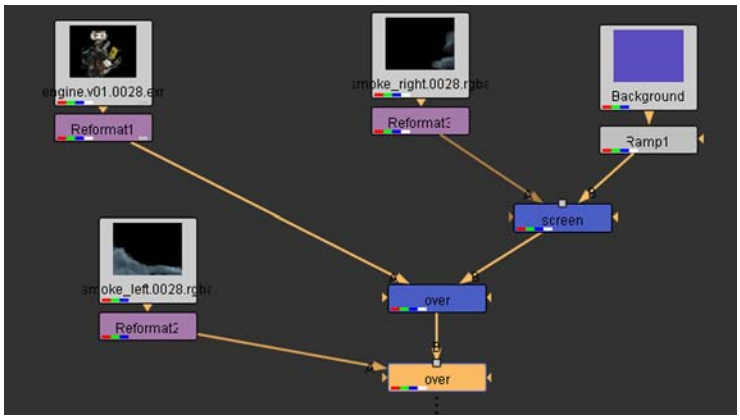
5. For the next layer, select the **Reformat3** node, attached to **smoke_right**. Then, hold down the **Shift** key and select the **Ramp1** node.
6. Press **M** to insert a Merge node and composite one image over the other. This composites the "smoke_right" image over the background.



7. The default compositing algorithm, "Over," isn't exactly what we need here. In the Merge2 control panel, click on the **operation** list and select **screen**.
8. In the Merge2 control panel, drag the **mix** slider and change its value to **0.30** to reduce the amount of the image supplied by the A input.



9. An additional Merge node is required. Select **Reformat2** for **smoke_left**. Hold down the **Shift** key and select the **Over** node (the first Merge node you inserted).



10. Press **M** to composite the two nodes. In the Merge3 control panel, change the **mix** slider to **0.75**.

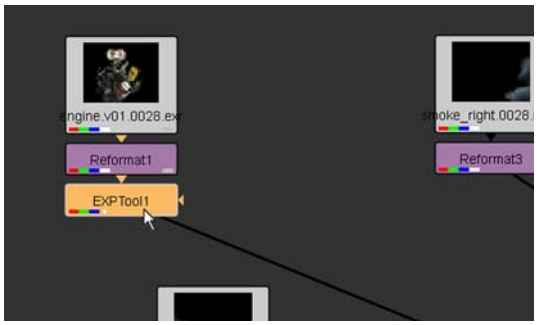
The result of your composite should look similar to the example shown below.

Figure 1-10: Result of the composite

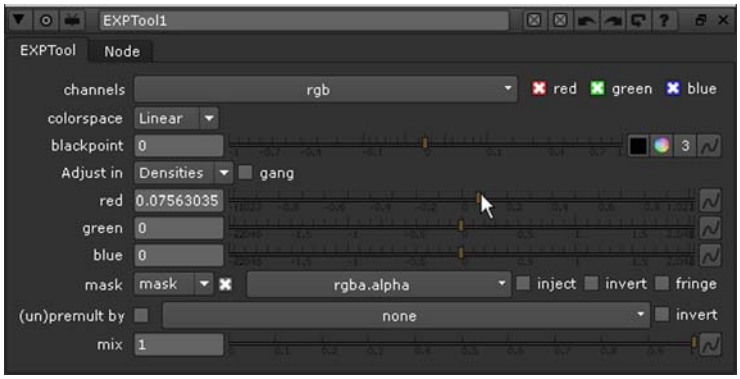
Colour-correcting Images

Colour-correction and filters can help you integrate the elements for a better composite. In our example, you want to limit the correction to the foreground element only, so you'll insert a colour correction node before the Merge nodes.

1. Select the **Reformat1** node. Then, right-click over the Node Graph and choose **Colour > Exposure**. This inserts the EXPTool1 node.



2. Suppose you want to adjust the value of the red colour channel. Move the mouse pointer over the Viewer window and press **R** to display the red channel.
3. In the EXPTool1 control panel, uncheck the box for **gang** sliders. This will allow you to adjust individual colour channels.



4. Drag the **red** slider to adjust the colour values. When you are finished, press **R** over the Viewer to display all channels.

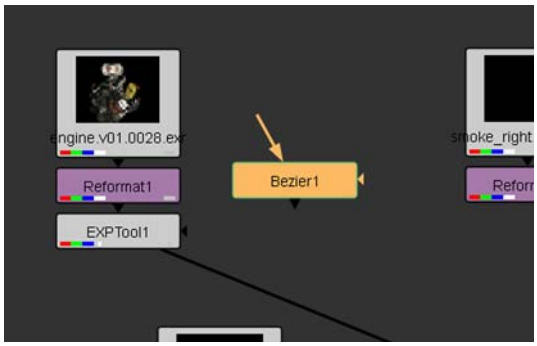
   The Exposure node worked as expected, but the result is less than spectacular. The colour change is too uniform. If only there were a way to limit—or, in fact, *mask*—the colour correction, perhaps we'd see a better composite. Hmm...

Masking Effects

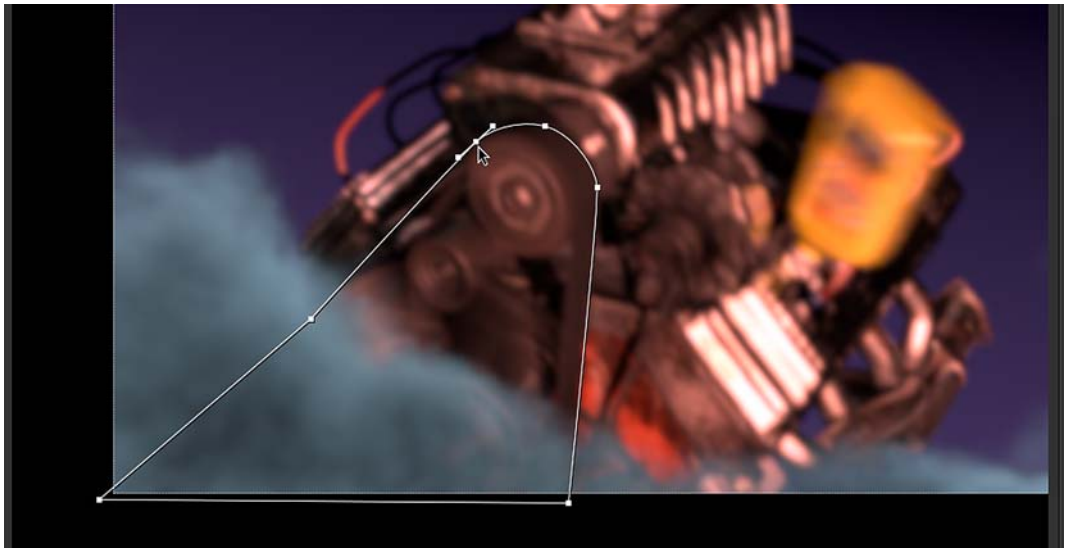
You can, indeed, apply masks to limit how each of these nodes affects the images. The following shows how to create a bezier mask to limit the colour-correction.

To create and apply a bezier mask

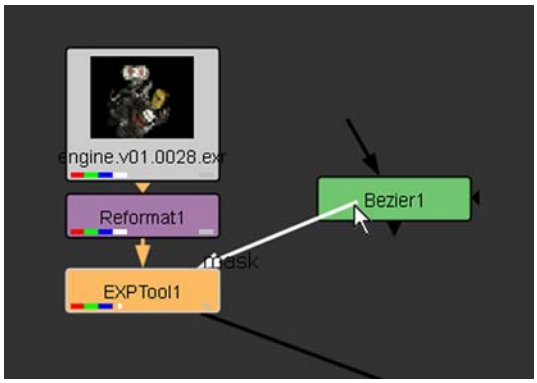
1. Click on a blank space in the node graph, so that nothing is selected in the node tree.
2. From the Toolbar, choose **Draw > Bezier** to insert a Bezier node.



3. Press both the **Ctrl** and **Alt** keys (Mac users press **Option+Command**) and click inside the Viewer window to draw a bezier shape over the image, like this:



4. To refine the shape, click on a point to select it and then drag to adjust its position.
5. To create sharp corners, select a point, right-click and choose **Cusp** from the pop-up menu.
6. To add points to the shape, simply hold down the **Ctrl** and **Alt** keys (or **Option+Command**) and click on the shape's outline.
7. When you're satisfied with the shape, drag the output connector from the **Bezier1** node to the mask port on the **EXPTool1** node.



In the EXPTool1 control panel, the mask channel option is now set to the **rgba.alpha** channel of the node that is connected to the **mask** input. In this case, this is the alpha channel of the Bezier1 node.

Creating Flipbook Previews

On the Viewer window, the timeline buttons let you play the project, but if you pay attention to the frames-per-second (FPS) field at the top of the Viewer window, you may notice that Nuke doesn't provide realtime playback. This is because Nuke renders on-the-fly to display images in the Viewer. It's fast, but also limited by the amount of memory and computer-processing power available to you.

The Flipbook feature provides better realtime preview, because it is prerendered for the FrameCycler viewer, included with Nuke. Keep in mind that the Flipbook feature will render a preview that matches the active resolution; if you're in proxy mode, for example, that's the resolution you'll get in the flipbook.

Note

The Flipbook feature renders temporary files in the directory you specified for "disk cache" under **Nuke > Preferences**. You'll also find an option there that allows you to limit the amount of disk space that the flipbook feature may use.

To generate a flipbook:

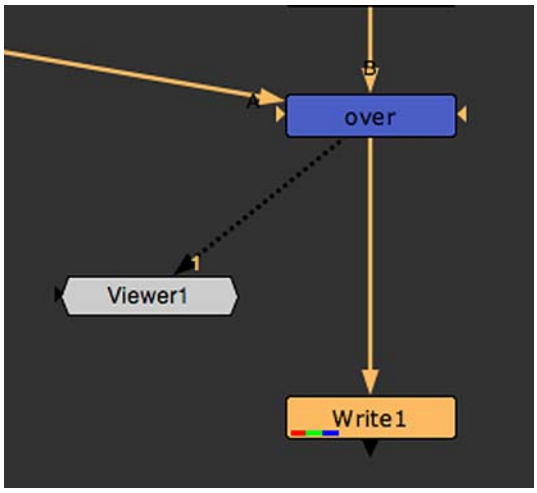
1. Select the **Over** node at the bottom of your node tree.
2. From the menu bar, choose **Render > Flipbook selected**.
3. Enter **1,28** as the number of frames to preview and click **OK**.
4. When the flipbook is ready to view, a copy of the Framecycler window will appear. Click the **Play** button to view the results.
5. Close the Framecycler window to return to your project.

Rendering Final Output

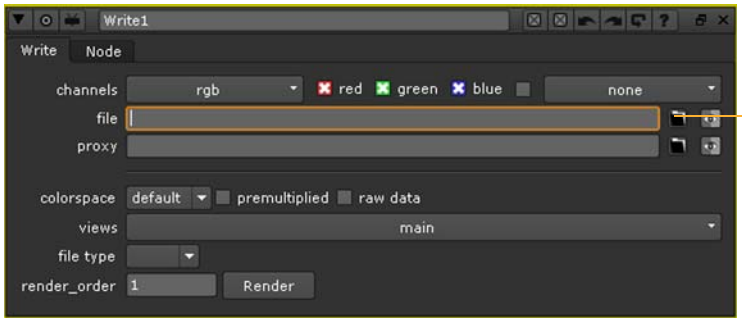
When you're ready to render the results of your composite, you insert a Write at the bottom of the node tree, and specify the pathname for the rendered images. Although we'll use just one here, you can place several Write nodes in your script, anywhere you like, to render output from different places in the tree. When the render order is important, use the **render order** option in the Write nodes to specify the order in which multiple renders should be executed.

To render the result of your composite:

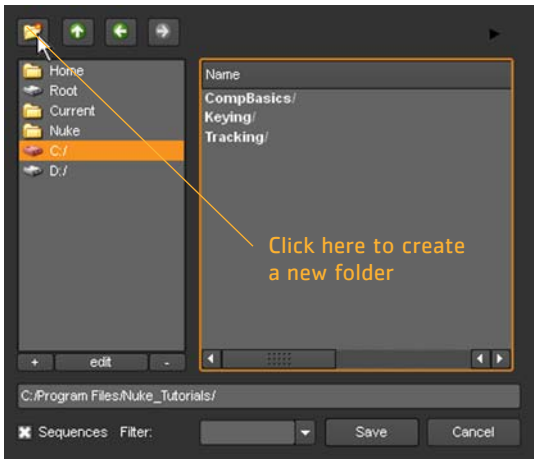
1. Select the last **Over** node at the bottom of the node tree.
2. Right-click and choose **Image > Write** to add a node for output.



3. In the control panel for the Write node, click the **file** folder icon.



4. Browse to the **Nuke_Tutorials** directory.



5. Click the "new folder" icon, in the upper-left corner of the browser, and type **Rendered** as the name for the new folder. Click **OK**.
6. Select the folder you just created.

   You should see the "Nuke_Tutorials/Rendered/" pathname displayed at the bottom of the browser.
7. At the end of the "Nuke_Tutorials/Rendered" pathname, type **first_comp.%04d.exr** as the name for the rendered image sequence, and then click **Save**.
8. Choose **Render > Render All** to render the images, or simply click the **Render** button inside the Write control panel.
9. Nuke prompts you to specify the frames to render. Enter **1,28** as the frame range and click **OK**.

A status window appears that shows the progress of your render. When the render is complete, you'll find the sequential images in the "Nuke_Tutorials/Rendered" directory. To check the results, simply insert a new Read node, point to the new image sequence, and then generate a flipbook with the Read node selected.

Using the Nuke Frame Number Variable

What's that weird "%04d" bit in the filename, you say? That's the variable that tells Nuke where to place the sequential numbers or frame numbers. You only type one name to represent the image sequence, but Nuke will create one image file for each frame in your shot.

So, in this case, you entered "first_comp.%04d.exr" but Nuke will render these files for frames 1 through 5: "first_comp.0001.exr," "first_comp.0002.exr," "first_comp.0003.exr," "first_comp.0004.exr," and "first_comp.0005.exr." You can change the variable—%03d, %02d, %05d—to change the number of padded digits for the frame numbers.

Image Formats

If you don't specify a file format inside the Write node control panel, Nuke uses the format specified by the filename extension you typed. For example, in this tutorial, you used the ".exr" extension to tell Nuke to save the images as OpenEXR files.

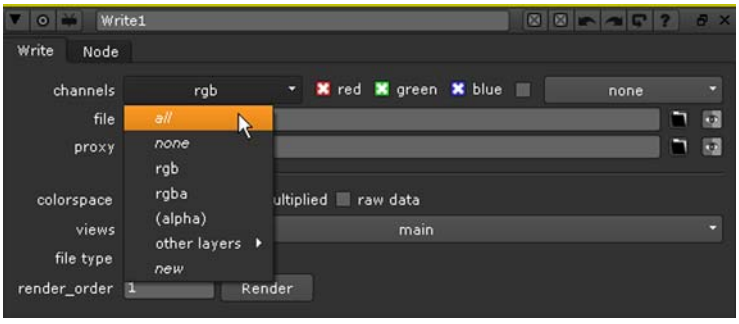
Rendering with the Active Resolution

When you execute a render or a flipbook, Nuke assumes you want to render the active resolution. When you're in full-res mode, for example, Nuke renders full-resolution images to disk. When you're in proxy mode, Nuke assumes you want to render the proxy resolution—defined in the Project Settings window—to the path and filename you specified as the **proxy** filename in the Write node. If the proxy field is empty or pointing to an invalid path, Nuke returns an error.

It's easy to toggle to proxy mode and then forget your images will be rendered in the lower resolution. Before you execute a render, it's always a good idea to check which resolution is active. In the Viewer, the label at the lower-right corner of your image will indicate whether you are in full-res or proxy mode.

Rendering Multiple Channels

When you insert a Write node, Nuke assumes that you need only the RGB channels in the final render. In many cases, this is acceptable because you won't need the alpha channel or other channels from the node tree when you deliver final shots to your clients. However, sometimes you need to render intermediate files—such as mattes, projection elements, or subcomps—and include all the channels in your node tree.



For example, rather than manage several elements for an animated character, you could combine the character animation, the lighting passes, alpha channel, and a depth mask in one image sequence on disk. This makes it easier to manage elements in the final composite and simplifies the artist's workflow.

To output all channels, change the Write node's Output list from **RGB** to **All Channels**, select the OpenEXR file format, and then execute the render. Currently the OpenEXR format (.exr) is the only file format that supports unlimited channels.

Epilogue

In this tutorial you setup a new project and created a simple composite. You learned how to use (or at least, locate) practically every Nuke window and tool, and you rendered out the result of your composite. You're finished! Go home!

Well... there might be a few more things you want to know. After this tutorial, you should feel comfortable with the Nuke user interface, so put on your explorer hat and review the other tutorials. There's no specific order from here, so look through the following pages until you find what interests you.

- End of Tutorial -

Tutorial 2: Tracking, Stabilising and Matchmoving

Every filmmaker knows the challenges of putting together a vision. You may not have the money to build post-apocalyptic Montreal, but you might have enough to create it in post. You may have brilliant performances by your actors—but not together in the same shot. Fortunately, you can composite the best takes. Your battle sequence with 5 A-list actors, 100,000 extras and 57 elephants, comes back from the lab with scratches on the negative. You can fix it. You can. A savvy production team knows how to leverage digital technology to make it possible, and Nuke's tracking tools are indispensable for these situations.

As you may know, tracking is the process of recording the location of features as they move through the scene. The result is stored as 2D coordinates on the image plane. Once you have the tracking data, you can use the movement to perform a variety of useful tasks, such as stabilising the footage, applying the movement to other elements in your composite, and improving the accuracy of roto mattes.

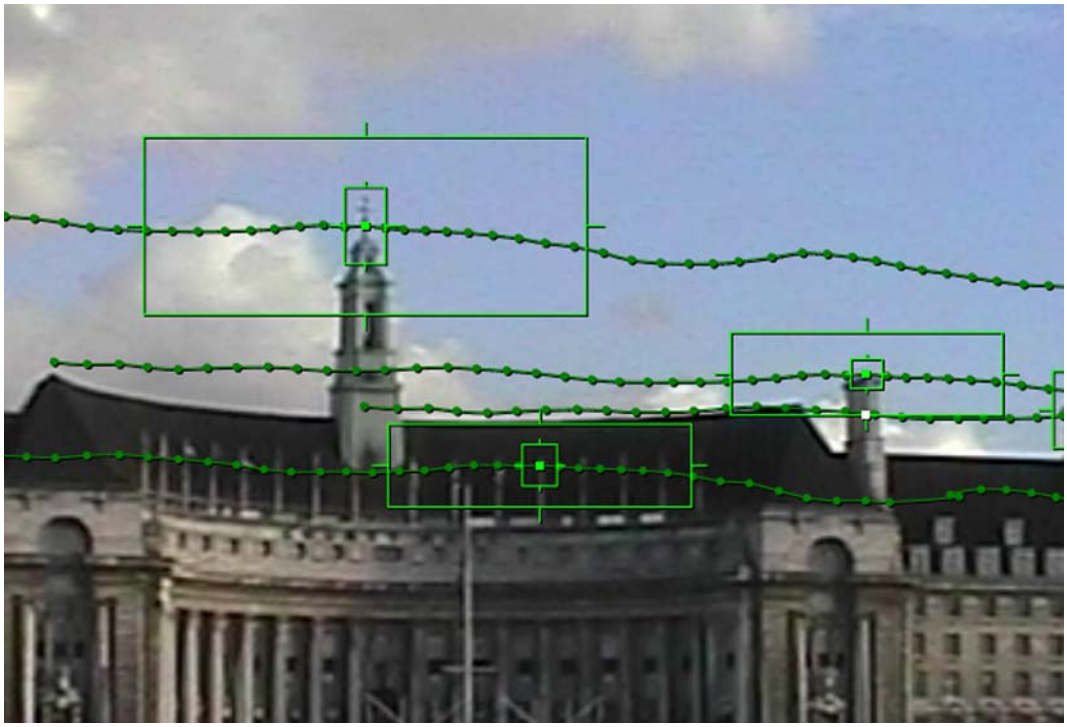


Figure 2-1: Tracking image features

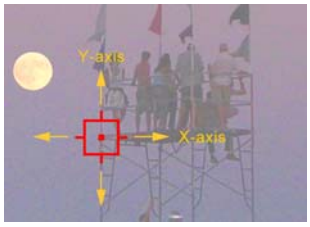
An important aspect of the tracking process involves carefully reviewing your footage *before* you attempt to track. Play your images several times—preferably with a flipbook—and look at the direction of movement for the features you want to track. Note potential problems with motion blur, obscuring objects, or frames where the features are hidden or move off screen.

Tip

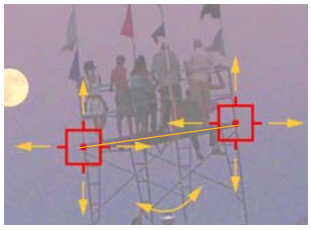
Nuke can often compensate for "problem footage," but tracking works best when you can identify distinct features throughout the length of the shot.

One-Point, Two-Point, Three-Point, Four

Before we get into the first example, let's review a few tracking concepts. You can track up to four distinct features or patterns with each Tracker node in Nuke. How do you decide whether to track one, two, or more features? It depends on what you want to do with the data and the level of accuracy you need in the result. Here are some general guidelines:



1 track: X and Y position only



2 tracks: X, Y, and Z-rotation

3 tracks: X, Y, Z-rotation, & scale

Figure 2-2: How many points do you need to track?

- **One-point tracking**. Track one feature's horizontal (x-axis) and vertical (y-axis) position, with little or no perspective change on the image. You can apply this information to move other elements in the composite or apply the inverse to stabilise the image.
- **Two-point tracking**. Track horizontal and vertical position for two features. The feature positions, relative to each other, indicate whether the image is rotating clockwise or counter-clockwise (z-axis rotation). In some cases, two tracking points is sufficient to calculate the scaling of the features, as well.
- **Three-point tracking**. Track horizontal and vertical position for three features. Provides all the benefits of two-point tracking with an additional set of tracking data for more accuracy on z-rotation and scaling.
- **Four-point tracking**. Again, all the benefits of the lesser tracks with an additional set of tracking data. Three-point is usually sufficient for most 2D tracking needs, but four-point makes it possible to distort and matchmove another element into the four points, or corners, of the features you track. That's why four-point tracking is typically called *cornerpin* tracking.

Open the Tutorial Project File

In this tutorial, you work from a project file that already includes the node trees. Each tree is setup for the examples that follow.

To open the project file:

1. Launch the Nuke application and choose **File > Open** from the menu bar.
2. In the file browser, navigate to your **Nuke_Tutorials/Tracking/** folder, select the **tracking_tutor.nk** project file and click **Open**.
3. It will show some nodes in error. Don't worry! It can't find the tutorial files. So before doing anything else you have to tell this script where to find these tutorial images, assuming you can remember where you put them. Double-click on the NoOp node in the top left corner of your node graph. It's called Tutorial_Path. Double clicking will bring up a property panel on the right. Enter the path to the tutorial files in the Tutorial Project

Directory. Use the file browser as that's often easier than typing it in. You should then see tutorial images appear.

4. Move the mouse pointer over the node graph, and press **F** to frame the entire contents of the project file.

The examples in this project file are grouped with coloured boxes, called *backdrops*, and each contains a node tree for the tutorial examples that follow.

Tip

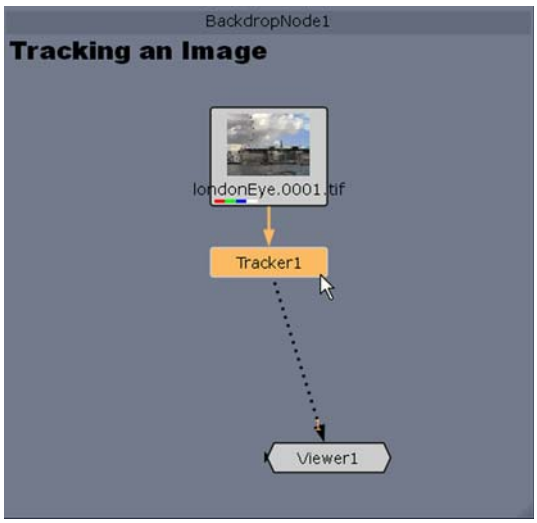
Backdrops let you organize groups of nodes, like those shown in this project file. Choose **Other > Backdrop** from the Toolbar. Drag the backdrop title bar to move it. Drag the backdrop corner to resize it. Any nodes surrounded by the borders of the backdrop will move with the backdrop when you drag its title bar.

Tracking a Single Feature

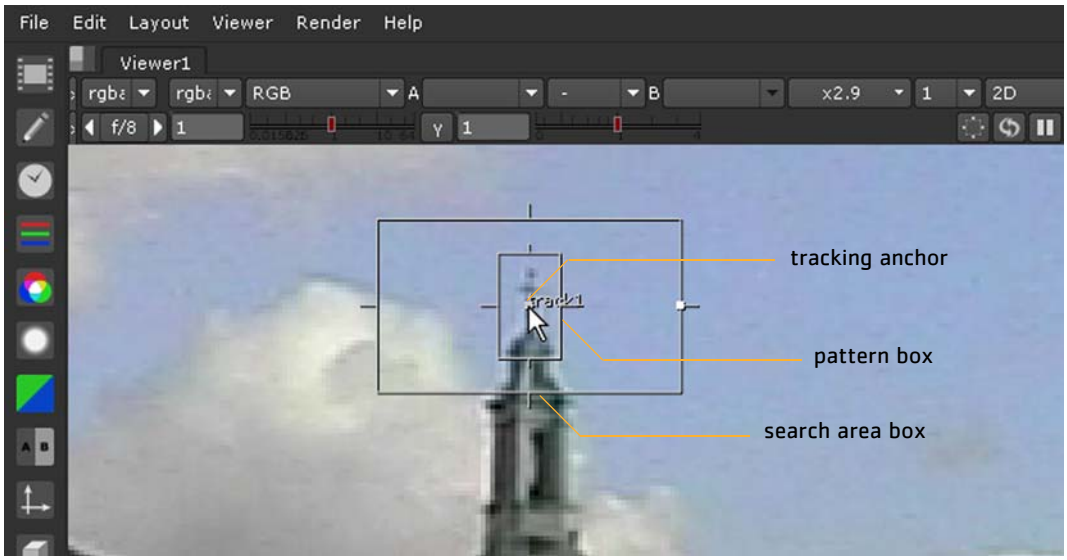
In this first example you'll learn how to track a single feature, which is the most basic 2D tracking operation. After you achieve a solid track for one feature, you can build on that and track other features as needed.

To track one feature (one-point tracking):

1. In the project workspace for the **tracking_tutor.nk** file, locate the node tree labelled "**Tracking an Image**."
2. Click on the **LondonEye** Read node to select it.
3. From the menu bar, choose **Render > Flipbook selected**. When the Framecycler window appears, play the flipbook several times to review the footage.
4. Look at the features in the image and notice the amount and direction of movement as the clip plays. When you're done, minimize the Framecycler window.
5. Choose **Transform > Tracker** from the Toolbar to attach a new Tracker node to the **LondonEye** Read node.



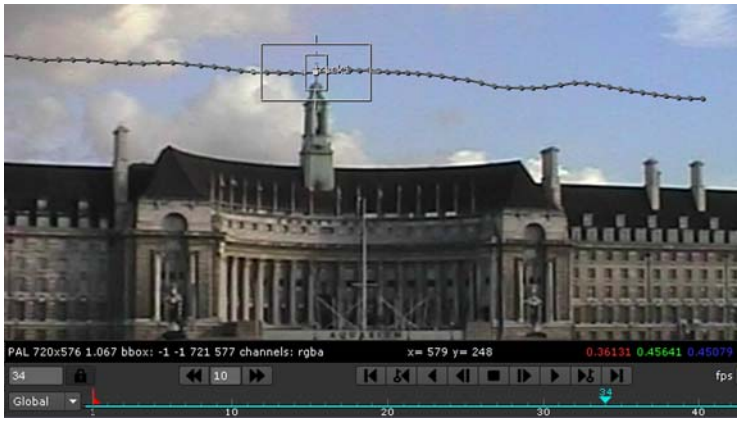
6. Connect the **Viewer1** node to the **Tracker1** node. Inside the Viewer, you'll see one tracking marker.
7. In the Viewer, scrub the time slider to frame 1, to make sure you're at the beginning of the shot.
8. Use the mouse to drag the tracking anchor over the tower spire, shown below:



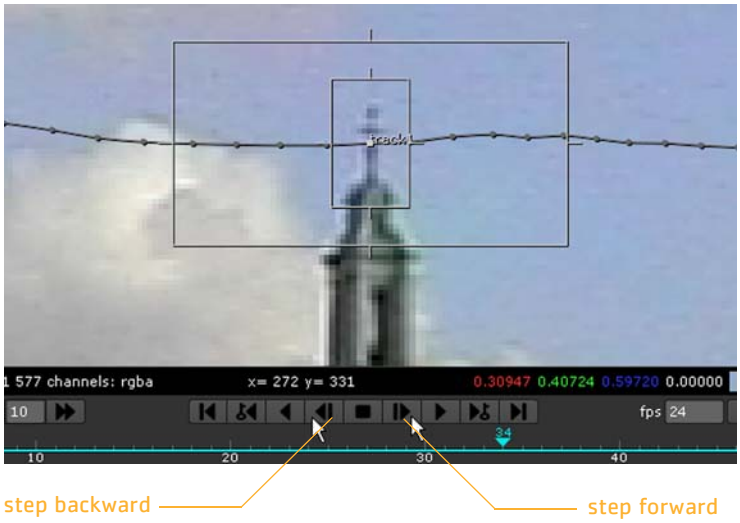
9. Click on the pattern box (inner box) of the tracking marker, and adjust its size to contain the feature.
10. Click the search area (outer box) of the tracking marker, and adjust its size to enclose the amount of space you think the feature may move between frames.

    Large search areas require more calculation time, so keep it as small as possible. However, when the search area is *too* small, the feature may move outside the box and you'll lose the track. If you aren't sure how large to make the search area, go back and review the flipbook of your image.
11. In the control panel, click the "track forward" button to generate the track.

When completed, you'll see the track curve with points that mark the key frame positions along the curve.



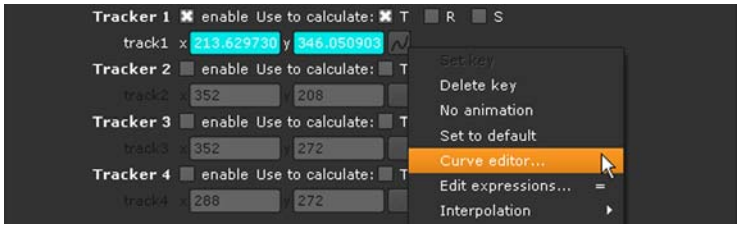
12. Use the "next frame" and "previous frame" buttons on the timeline to step through the timeline and verify the accuracy of the track.



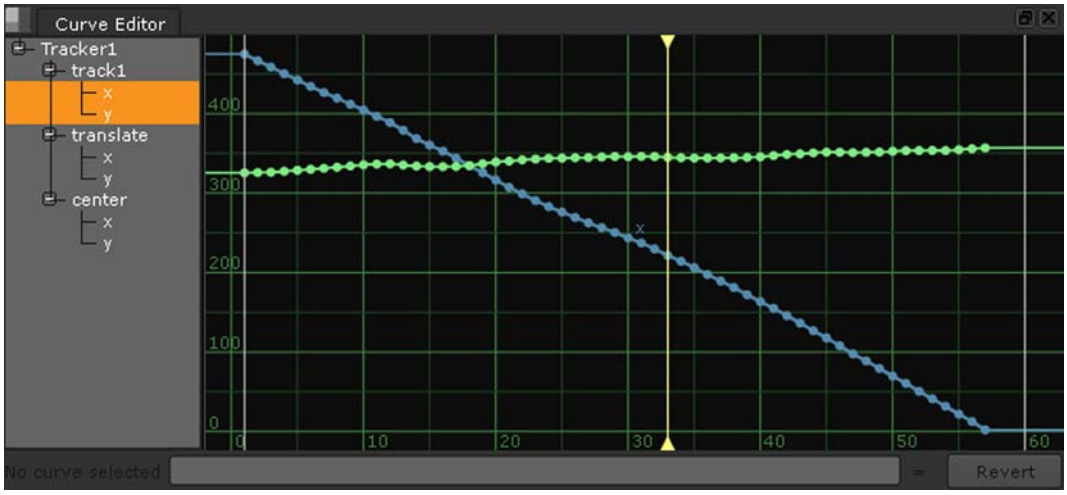
The track is fairly solid in this example. However, some images don't track as easily as this one. When you need to edit track data, what do you do? You open the Curve Editor of course! Let's assume you need to smooth the points of this track.

To edit track data:

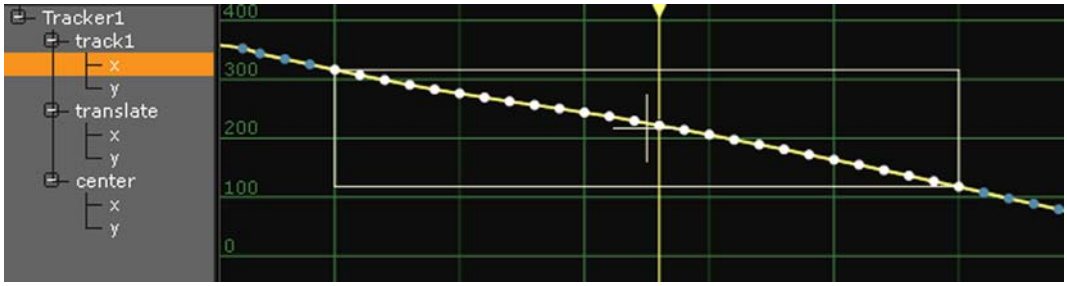
1. In the tracker control panel, click the animation button next to the **Tracker 1** parameter, and choose **Curve editor** from the pop-up menu.


Tracker 1 enable Use to calculate: T R S
track1 x 213.629730 y 346.050903
Tracker 2 enable Use to calculate: T
Tracker 3 enable Use to calculate: T
Tracker 4 enable Use to calculate: T
Set key
Delete key
No animation
Set to default
Curve editor...
Edit expressions...
Interpolation

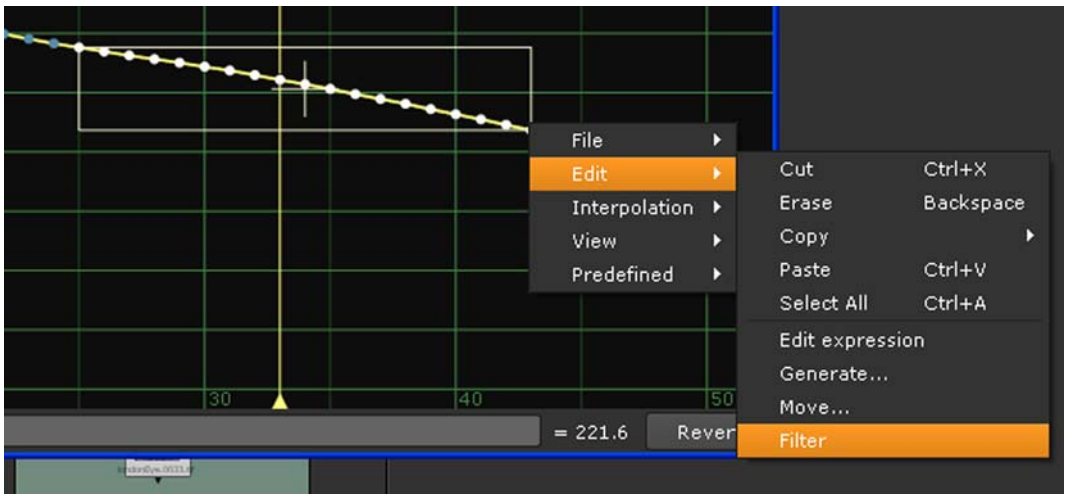
2. Click the items in the Curve Editor outline and you'll see values recorded for each of the parameters during the tracking process.



3. Hold down the **Shift** key and click the **X** and **Y** curves, under **track1**, to select both curves at once and press the **F** key to "frame" the curves.
   - To adjust a value, select a point and drag it up or down.
   - To change the frame for a particular point, select it, hold down the Ctrl key and drag the point left or right.
4. Let's assume you want to smooth a curve by applying a filter to the values. Draw a marquee—drag while pressing the left mouse button—around a section of the curve to select multiple points.



5. Click the right mouse button to display a pop-up menu of Curve Editor options. Choose **Edit > Filter** and enter **2** as the number of times to filter the key frames.



Nuke averages the location of each point based on the values of the surrounding points, and this helps to smooth the curve.

6. Close the Curve Editor window and then play the result in the Viewer.

Those are the basics for tracking and editing the results for a single feature. In the next example, we'll make it a little harder—tracking a feature that moves out of view.

Tracking Obscured Features

At the end of the previous example, you may have noticed the track was dropped at frame 58 when the feature moved off the screen. Disappointing? Yes. A disaster? Probably not. When features move out of frame, or become obscured by other elements in the image, you can use the track offset feature to pass the tracking operation to another feature in the image. Nuke then attempts to continue the track along its current course.

To track a feature that moves off screen:

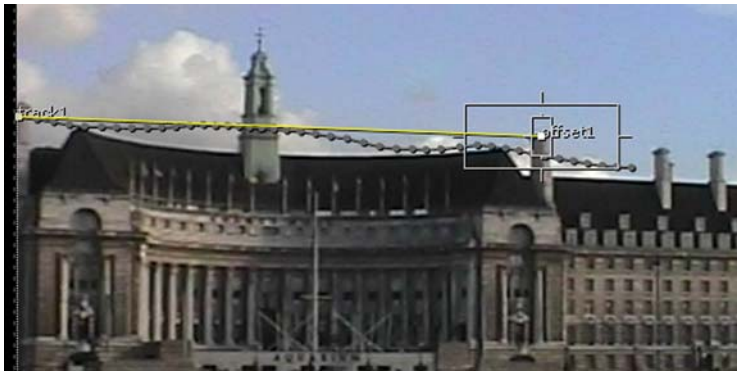
1. In the project workspace, locate the node tree "**Tracking Obscured Features**."
2. Double-click the **Tracker2** node to open its control panel. This node tracks one of the chimneys in the clip you used from the previous example.
3. Attach a Viewer to the **Tracker2** node and scrub the timeline until you see the tracked feature move out of frame.

As you can see, track1 accurately tracks its feature through most of the clip—until it moves off the screen at frame 44. This is where the problem starts.

4. Press the plus key (**+**) on your keyboard a few times to zoom in on the Viewer.

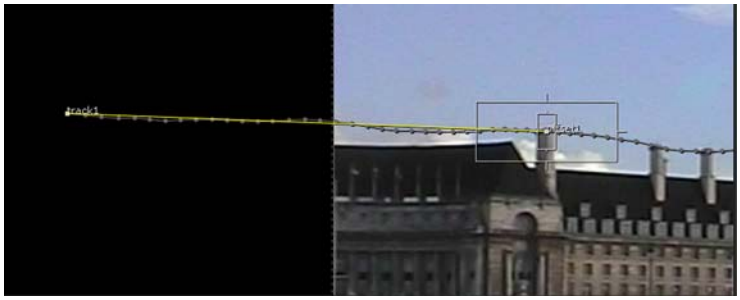
   You want to select an alternate feature that stays in view during the length of the clip.

5. At frame 44, press the **Ctrl** (or **Command**) key and drag the **track1** point to the first chimney at the right.



   The "offset1" label appears with a line connecting the new feature to the original feature.

6. In the **Tracker2** control panel, press the "track forward" button and Nuke continues the track off the screen.



   Why is this cool? Because you can now use the track data to matchmove an element—a trail of chimney smoke, for example—that will lock to the feature even after it moves off the screen.

7. The track is now complete, so you can click the **clear offset** button in the Tracker2 control panel.
8. Uncheck the **enable** box for **track1** to prevent it from being recalculated.
9. Before you continue, close all Tracker control panels that are currently open.

The offset doesn't change the track location. Instead, it allows Nuke to continue the track with the assumption that the offset feature remains at the same relative distance to the original feature. Later in this chapter, you'll see how to use this tracking data to composite another element to match the background plate.

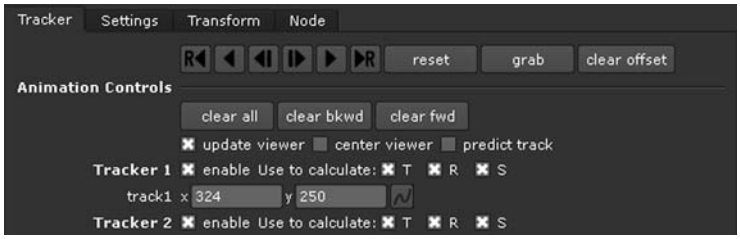
Stabilising Elements

Stabilisation is the process of removing motion—camera-shake, for example—and locking down the element for your composite. A one-point track provides enough information to sta-

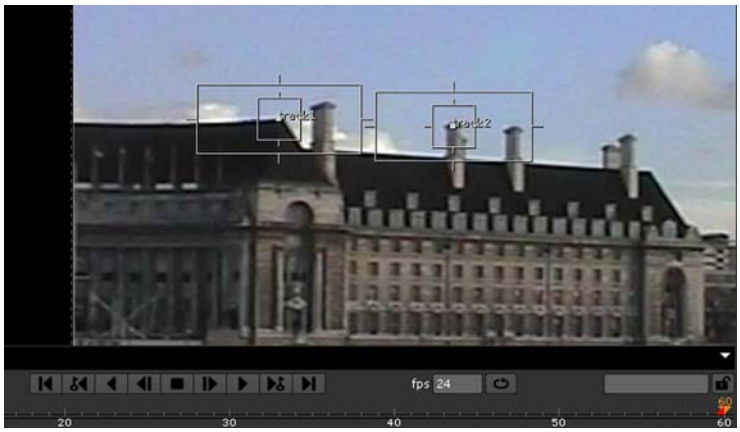
bilise horizontal and vertical motion along the image plane. A two-point track lets you stabilise horizontal and vertical motion, and remove rotation in the image, as well.

To track and stabilise:

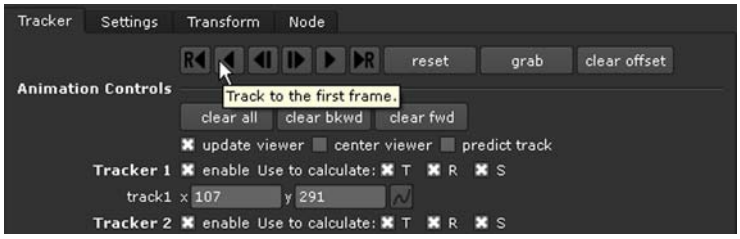
1. Locate the node tree labelled "**Stabilizing Elements**."
2. You'll see a copy of the same **LondonEye** Read node that we've been using for the other examples. Click on it to select it.
3. Choose **Transform > Tracker** and then attach a Viewer to the new **Tracker3** node.
4. In the control panel for Tracker3, check the boxes to **enable Tracker1** and **Tracker2**.
5. For each track, check the boxes for **T** (translate), **R** (rotate) and **S** (scale).



6. In the Viewer, scrub to the end of the timeline and adjust the size and position of each tracking marker for the features shown below:

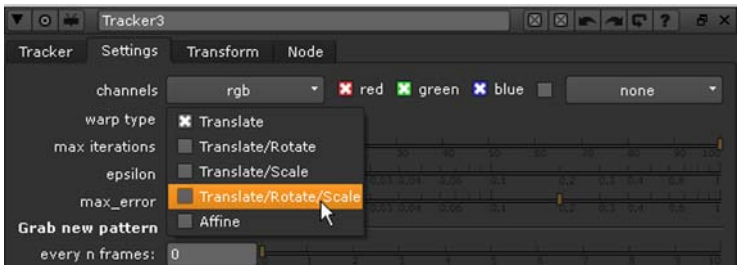


7. In the control panel, click the "track backward" button to generate the tracks.

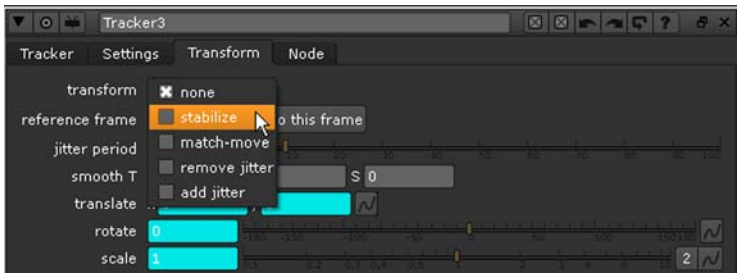


Now you have the position data for two tracks, and you can use this information to remove the unwanted movement in the image.

8. In the Tracker3 control panel, click the **Settings** tab and choose **Translate/Rotate/Scale** from the **warp type** list.



9. Click the **Transform** tab and choose **stabilize** from the **transform** list.



10. In the Viewer, click the "play forward" button and review the results.

As the clip plays, you'll see the features remain locked to the same position within the compositing frame.

Tip

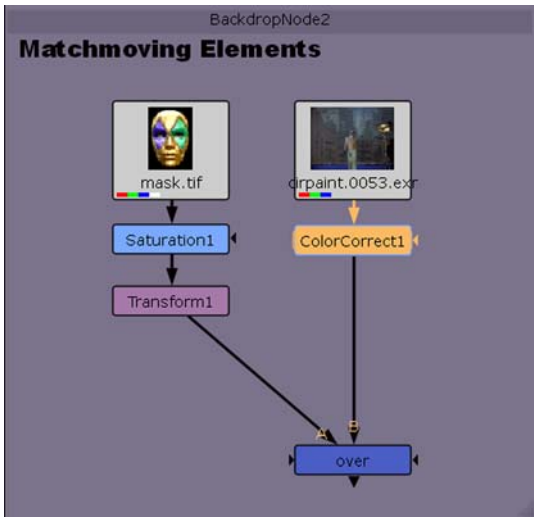
After you track and stabilise footage, you can add a **Transform** > **Transform** node after the Tracker3 node to adjust the position and the rotation of the stabilised image for a final composite.

Matchmoving Elements

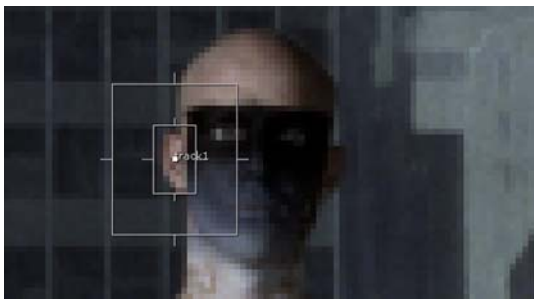
Matchmoving is the opposite of stabilisation. The intent is to record and use the motion in an image and apply it to another element. In the following example, you'll use the tracker to matchmove and composite a mask image onto the performer in a background plate.

To matchmove an element:

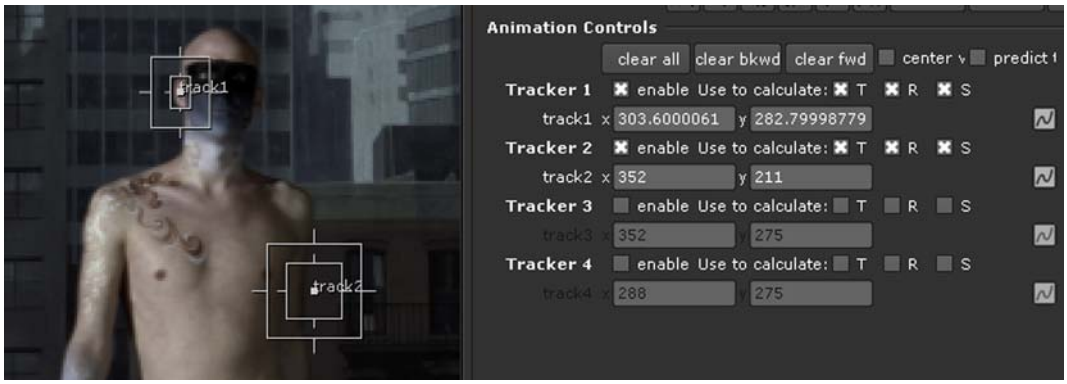
1. Find the node tree labelled "**Matchmoving Elements**."
2. Drag the time slider to the beginning of the timeline. Select the **ColorCorrect1** node and then choose **Transform > Tracker**.



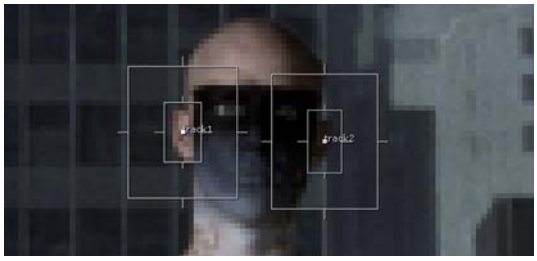
3. Attach a Viewer to the new **Tracker4** node and position the **track1** marker over the performer's right ear.



4. Adjust the size of the pattern box and the search area as shown.
5. In the control panel, next to Tracker1, check the boxes for **T** (translate), **R** (rotate) and **S** (scale).
6. Click the **enable** box for **Track 2** to activate the controls for an additional track. Check the boxes for **T** (translate), **R** (rotate) and **S** (scale) on this track, also.
7. Press the minus key (**-**) over the Viewer to zoom out, and you'll see a second tracking marker appears.

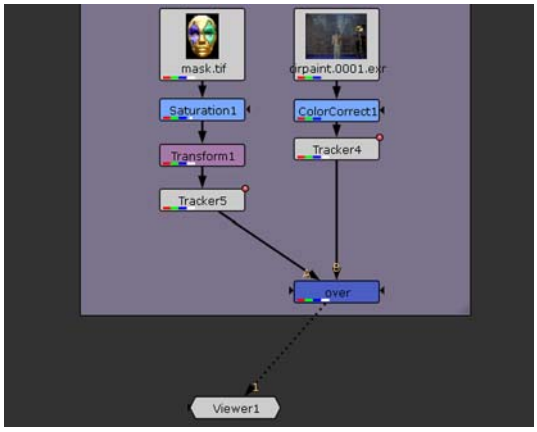


8. Adjust the size and position of the second tracking marker, as shown below.

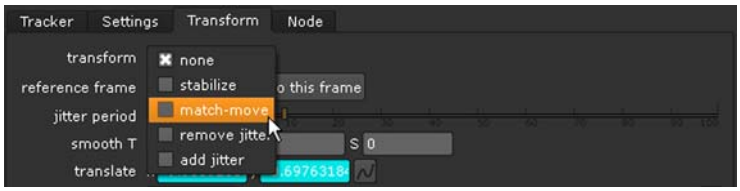


9. In the control panel for Tracker4, click the "track forward" button to generate the tracks.

   After you get a solid track on the performer, you need to make a copy of the Tracker4 node to create the matchmove.
10. Select the **Tracker4** node and press **Ctrl+C** to copy it.
11. Select the **Transform1** node and press **Ctrl+V** to paste the Tracker node copy (Tracker5).
12. Connect the Viewer to the **Over** node. Your node tree should now look similar to this:



13. In the Tracker5 control panel, click the **Transform** tab and choose **match-move**. Then close the Tracker5 control panel.



14. Click the "play forward" button in the Viewer or render a flipbook and you should see the Mardi Gras mask transform to match the movement of the performer.

If you see jitter in the movement, you can edit the track data in the Curve Editor to smooth out the data. You can also add a small value to the de-jitter parameter or add values to the smooth T, R, and S parameters on the Transform tab to filter the tracks.

Epilogue

In this tutorial, you worked with several examples for the Tracking node. You learned how to record the locations for multiple features and you applied the tracking data for other tasks in the composite, such as stabilisation and matchmoving.

- End of Tutorial -

Tutorial 3: Keying and Mattes

Keying is one of those fundamental compositing skills. You can't composite anything until you have mattes pulled for the elements you want to layer together. It's nice to say you could just push a button to complete this task, but as you probably know, one keying operation seldom produces an acceptable matte. Image quality, lighting conditions, subject motion, colours—even camera moves—affect the steps required to get a clean matte for your composite.

Figure 3-1: Keying Footage in Nuke

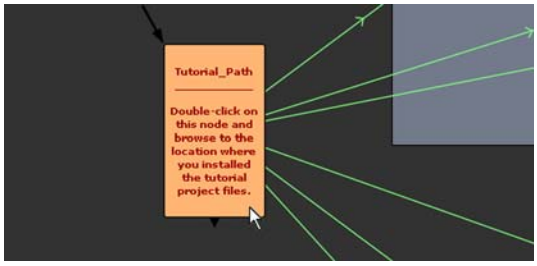
So how do you get a clean matte in Nuke? The best approach is to understand the strengths of each keying tool and combine them as needed. This tutorial shows how to pull keys in Nuke and how to layer the results with channel operations, merge nodes, and rotoshapes.

Open the Tutorial Project File

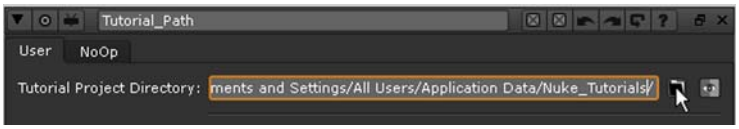
The project file for this tutorial includes several node trees for the keying operations described in this chapter.

To open the project file:

1. Launch the Nuke application and choose **File > Open** from the menu bar.
2. In the file browser, navigate to your **Nuke_Tutorials/Keying/** folder, select the **keying_tutor.nk** project file and click **Open**.
3. Double-click on the **Tutorial_Path** node, located on the left side of the script, to open its control panel.



4. In the **Tutorial_Path** control panel, click the "file folder" button. Browse to the location where you installed the tutorial project files, and then click **Open** to select the location.



   After you select the correct path, the error messages should clear from the Read nodes, and the thumbnails in the script will update with the correct images.

5. Close the **Tutorial_Path** control panel. Then, choose **File > Save As** to save a copy of the project file.
6. Move the mouse pointer over the node graph, and press **F** to frame the entire contents of the project file.

   The green arrows (lines) show the links between the Tutorial_Path node and the Read nodes.

7. If you wish, press **Alt+E** to hide the expression arrows.

   The Tutorial_Path node saves the location of the project files on your computer, so you don't need to repeat this for future sessions.

Figure 3-2: Node trees in the keying_tutor.nk project file

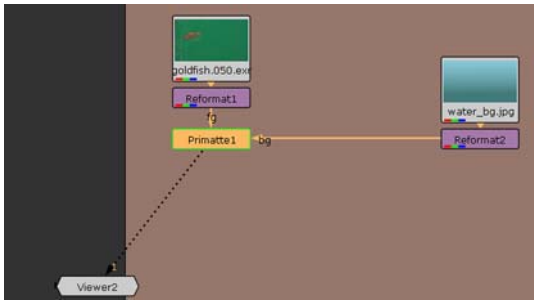
Keying with Primatte

The Primatte keyer includes a quick "Auto-Compute" option that evaluates your image and determines a good baseline key. From there, you can easily tweak the settings and generate an acceptable matte.

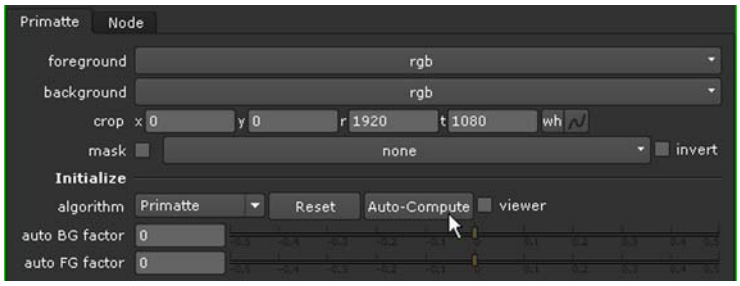
The two examples in this section show how to pull a key with the Auto-Compute option (method 1), and also how to manually sample a colour from the screen background and build your key from there (method 2).

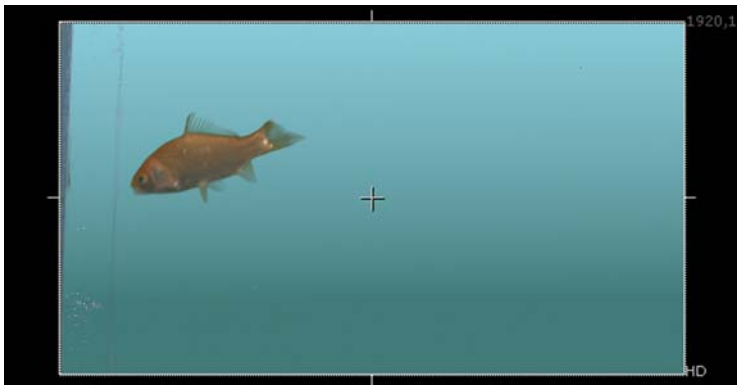
To pull a key with Primatte (method 1):

1. In the project file, locate the node tree labeled "**Keying with Primatte**," and make sure a Viewer is attached to the **Reformat1** node.
2. Choose **Keyer > Primatte** to insert the keyer between the foreground image and the Viewer.



3. Drag the **bg** connector from **Primatte1** to the **Reformat2** node, which supplies the background image for this example. The **fg** connector should be attached to **Reformat1**.
4. Move the time slider to frame **50**, and click the **Auto-Compute** button inside the **Primatte1** control panel.





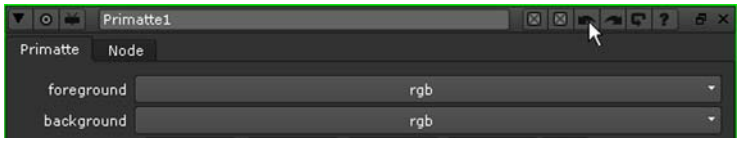
That's it. You're done... well, nearly done. We need a "free-floating" goldfish, but the reflections in the aquarium glass clearly indicate "captivity."

A garbage matte will easily remove the reflections, and you'll learn how to do that later in the section on rotoscoping. For now, let's keep working with Primatte.

As you've seen, Primatte's auto-compute option can quickly pull keys on certain images. However, you should also know how to pull and tweak keys manually. You might, for example, need more control over the transparency of the fins on the goldfish.

To pull a key with Primatte (method 2):

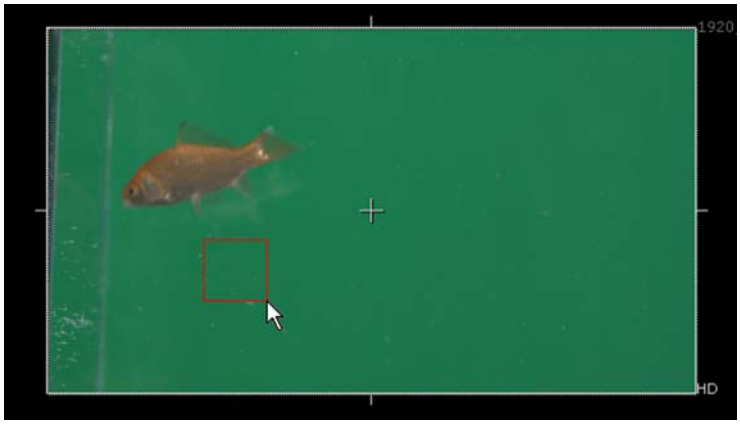
1. Continuing from the previous example, open the **Primatte1** control panel.
2. Click the "undo" button at the top of the control panel to step back to the previous state of the **Primatte1** node. Or, you can also delete the current **Primatte1** node and insert a new one.



3. Scroll down through the Primatte options and set the keying **operation** to **Select BG Colour**.

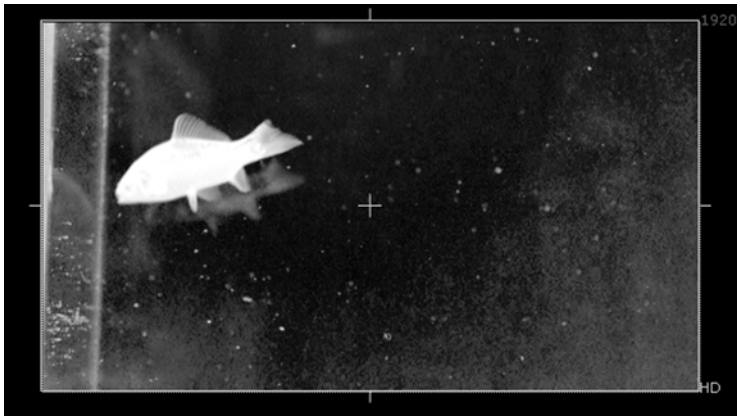
4. The **current colour** chip should display the eyedropper icon. If it doesn't, click on the colour chip to toggle the eyedropper.

5. Hold down the **Ctrl+Shift** keys (Mac users, hold down **Command+Shift**) and drag—or scrub—over a portion of the greenscreen in the image displayed in the Viewer.



This returns an average colour-pick of the sampled pixels. If you want a colour pick from a single pixel, press **Ctrl** or **Command** and click once over the greenscreen. After you pick, you can clear the red square by **Ctrl-** or **Command-**clicking again.

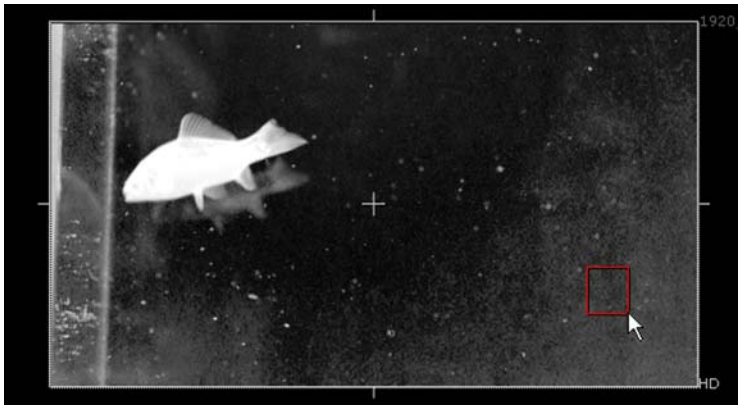
6. Press **A** over the Viewer to toggle to the alpha channel display. Looks like the aquarium is not as clean as we thought. Our colour pick gave us a fairly noisy key, so let's clean it up.



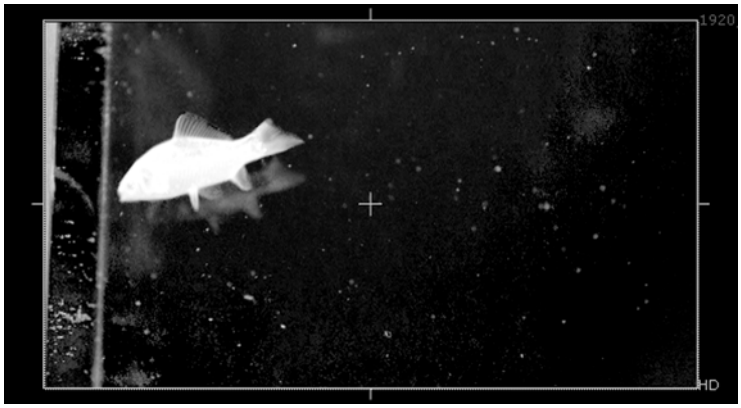
Now you'll sample a few areas of the image to "push" selected pixels to one of three areas: the transparent matte, the opaque subject, or the semi-transparent part of the matte.

7. In the **Primatte1** control panel, change the keying **operation** to **Clean BG Noise**.

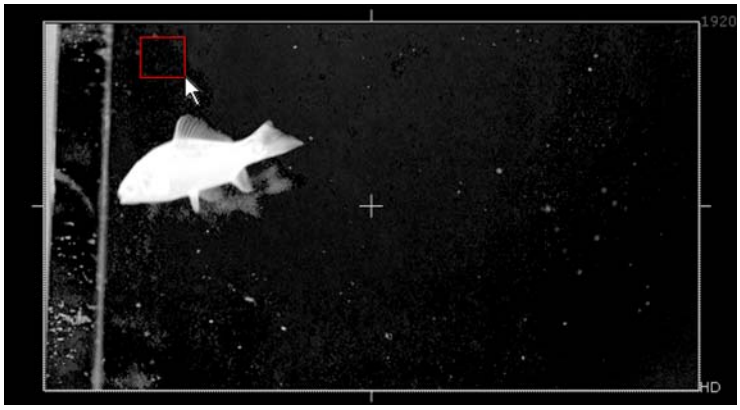
8. Press **Ctrl+Shift** or **Command+Shift** and drag a small square over the dark area in the lower-right corner of the image.



This second colour sample cleans the background by "pushing" the selected pixels into the transparent area of the matte. You probably need a few more samples to get a better key.

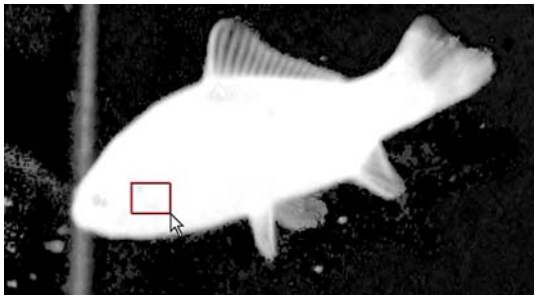


9. Scrub a few small areas in the background, focusing on the grey pixels until the matte improves.



The background doesn't need to be solid black. We're just trying to get a good separation between our foreground subject and the greenscreen background.

10. Change the keying **operation** to **Clean FG Noise**. This time, sample areas of grey pixels inside the goldfish. One or two small samples should be enough. The colour pick pushes the selected pixels to the opaque part of the matte.



You want to keep the grey pixels inside the fins to retain a semi-transparent matte in these areas. If you go too far, you can always press the undo button in the control panel to step back to the previous action.

11. Press **A** again over the Viewer to toggle to all colour channels. Your image should look similar to the example shown below. You may see some detail dropping out from the fins.

12. Change the keying **operation** to **Restore Detail**, and scrub over the fins to bring back some of the edge detail.

You may get different results than those shown here, depending on the pixel values you sample from the image.

Use Restore Detail to push the selected pixels back toward the opaque part of the matte. Use the Make FG Transparent operation to fine-tune the semi-transparent area.

You could go back and forth, between cleaning the background and foreground, but this usually produces a matte with "crunchy" edges. The goal is to find the balance between foreground and background that produces an acceptable matte for your subject.

Later in this chapter, you'll use the rotoscoping tools to clean-up this matte and combine this with the image from the next example.

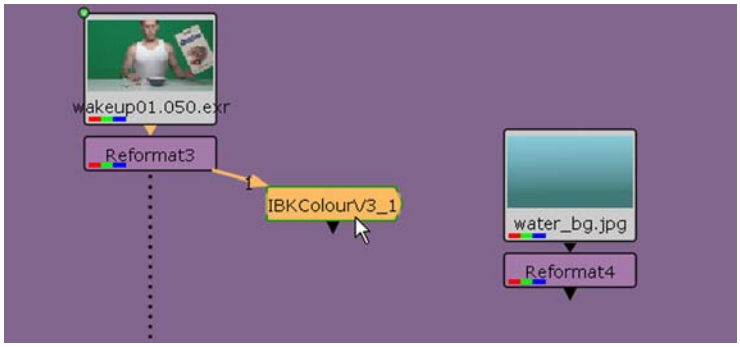
Image-based Keying

Many keying tools, like Primatte, use a colour-pick as the baseline for the matte extraction process and then require the artist to tweak the matte from that baseline. Nuke's image-based keyer (IBK) uses the pixel values of the compositing images, instead of a colour-pick, to generate the best matte for the image you want to extract. It works by generating a processed screen image that preserves the colour variations of the blue- or greenscreen and using this - rather than a single colour - to pull the key. This generally gives good results and speeds up the keying process when working with uneven blue- or greenscreens.

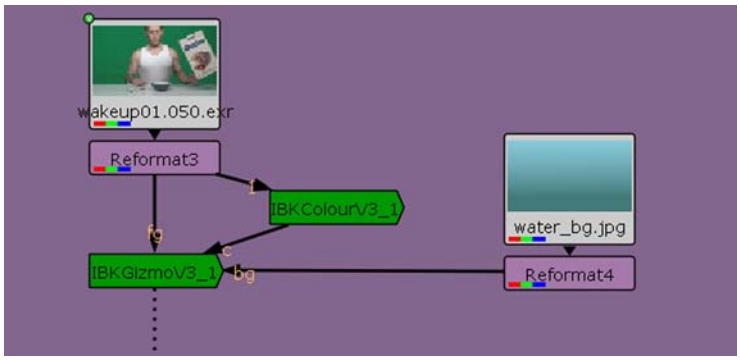
Image-based keying requires two nodes in Nuke. First, you insert an IBKColour node to process the screen image, which is preset to work with either greenscreen or bluescreen. This node generates the processed screen image that preserves the colour variations in your blue- or greenscreen. Then, you insert an IBKGizmo node to generate the matte using the processed screen image, the original image, and also the background image for the composite.

To pull a key with IBK:

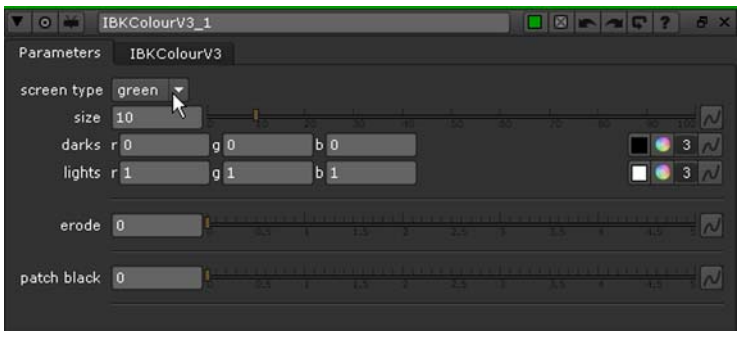
1. In the **keying_tutor.nk** project file, locate the node tree labelled, "**Image-based Keying**".
2. Right-click over the **Reformat3** node and choose **Keyer > IBKColour** from the pop-up menu. Drag the **IBKColourV3_1** node to the right.



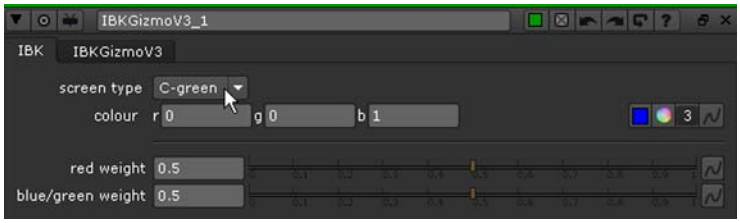
3. Click an empty spot in the node graph to deselect all nodes. Then, choose **Keyer > IBKGizmo** from the pop-up menu.
4. From **IBKGizmoV3_01** node, connect **fg** (foreground) to the **Reformat3** node. Connect **c** (colour screen) to the **IBKColourV3_1** node.
5. Connect **bg** from **IBKGizmoV3_1** to the **Reformat4** node, which supplies the background for the comp.
6. Connect the Viewer to the **IBKGizmoV3_1** node, and your node tree should look similar to this:



7. Open the control panel for **IBKColourV3_1** and change the **screen type** to **green**.



8. Open the control panel for **IBKGizmoV3_1**, and change its **screen type** to **C-green**.

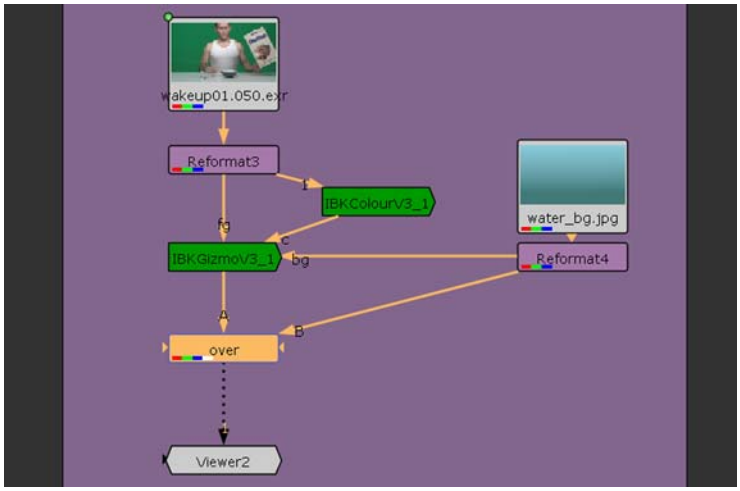


You should see an acceptable matte, shown in the screen capture below, on frame 50.

This is a very good start for this image.

9. Connect the Viewer to the **IBKColourV3_1** node. You'll see the processed screen image, which is essentially a Gaussian-filtered high-contrast key.

10. Choose **Merge > Merge** (or press **M** over the node graph) to insert an **over** node.



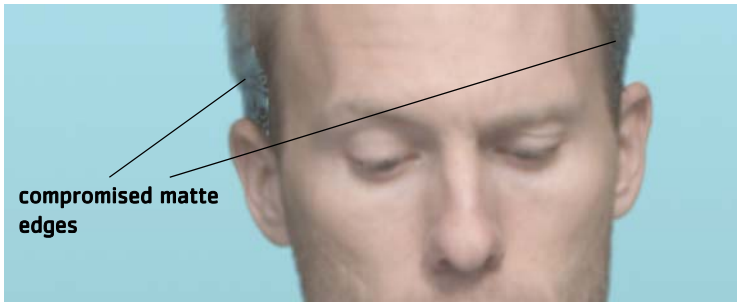
11. Connect **IBKGizmoV3_1** to the **A** input of the **over** node. Then connect the **B** input to the **Reformat4** node.

The colour of this greenscreen is completely out of the region of the acceptable industry standard, but IBK does a good job anyway, by smoothing the screen and using the result to essentially create a difference matte with the foreground.

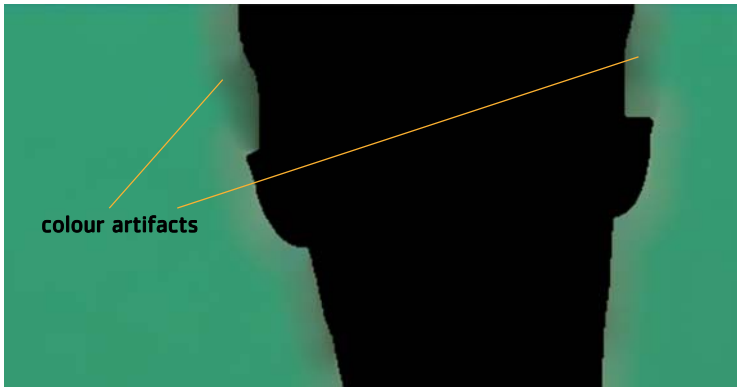
Tip

IBK has presets for green and blue screens, but you can also do a colour-pick for any screen colour inside the IBKGizmo node.

If you zoom-in on the image, you'll see small areas near the subject's hair, where the matte is compromised.

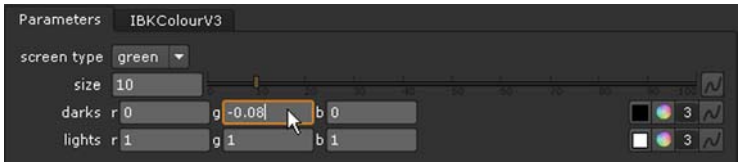


12. Connect the Viewer to **IBKColourV3_1** and you'll see colour artifacts around the edges of the matte.



When you look at the smoothed screen produced by IBKColour, you should see only values of your screen colour and black.

13. In the **IBKColourV3_1** control panel, lower the **darks**, **g** (green) setting to **-0.08**. (If you were keying a bluescreen image, you would lower the "b" value for "darks.").

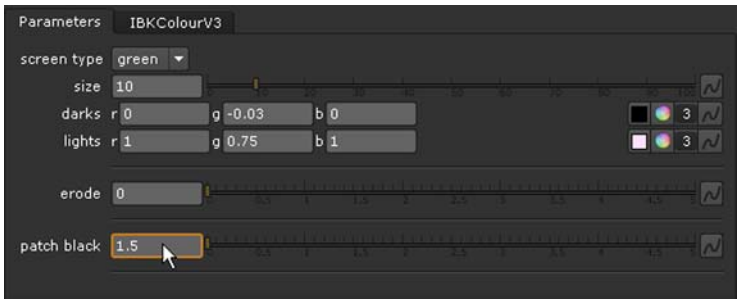


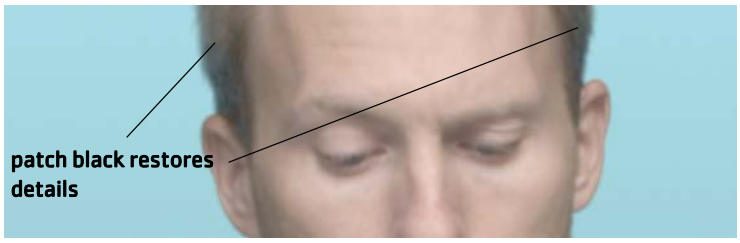
This fixes most of the problems at the hairline, but destroys the good key for the lower portion of the image.

Tip

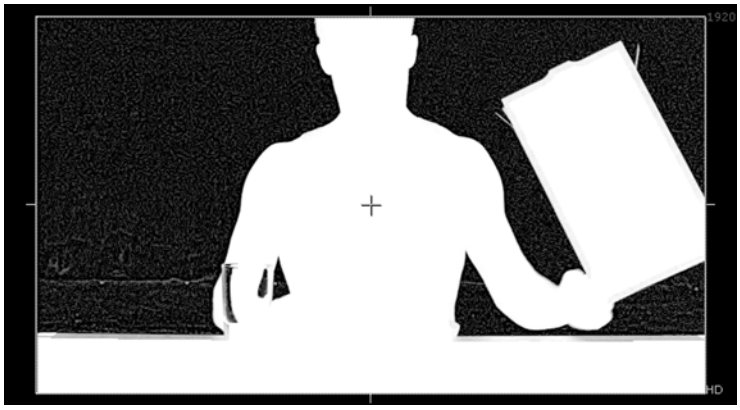
The artifacts in the IBKColour image appear as specific colour shades: light green, dark green, light red, dark red, light blue, and dark blue. To remove these, simply adjust the appropriate controls for the artifacts you want to remove: lights/g, darks/g, lights/r, darks/r, lights/b, and darks/b.

14. In **IBKColourV3_1**, raise the **darks**, **g** value to **-0.03**. Then change the **lights**, **g** value to **0.75**. This corrects the artifacts of the screen image.
15. Now, change the **patch black** setting to **1.5** to restore the edge detail of the hairline.





16. Connect the Viewer to the **IBKGizmoV3_1** node. Press **A** and you'll see the current alpha channel produced by the IBK system.



The displayed alpha image shown is correct for the IBK. If the intensity of the noise in your alpha channel is greater than the example show above, you may need to adjust—in very small increments—the **dark** and **light** values for the colour channels in the IBKColour node.

17. Press **A** again over the Viewer to toggle back to display all colour channels, and scrub through the timeline to check the matte at various points in the clip.

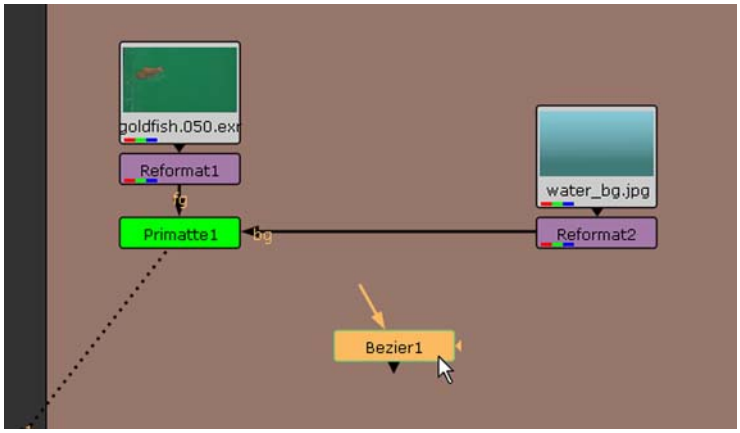
18. If you haven't already done so, save your project under a new file name to save the changes you've made to project.

Rotoscoping

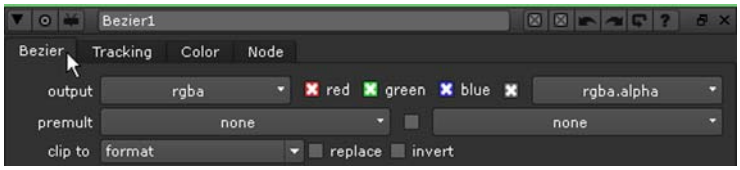
In this example, we'll return to our first keying example to apply a garbage matte and clean-up the aquarium image.

To draw a garbage matte:

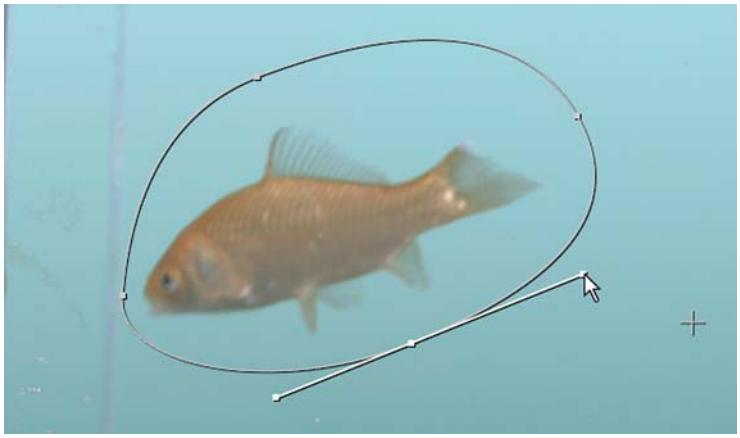
1. Go back to the node tree from the first example, and connect the Viewer to the **Primatte1** node. Drag the time slider to frame **50**.
2. Click an empty spot on the node graph to deselect all nodes, and choose **Draw > Bezier** from the pop-up menu.



3. At this point, you don't need to connect the **Bezier1** node to anything, but its control panel must be open, and the first tab, **Bezier**, should be active.



4. Inside the Viewer, you'll see the goldfish image. While pressing both **Ctrl** and **Alt** (Mac users, press **Option** and **Command**), click four points around the goldfish to create a rotoshape.



Tip

As long as the Bezier1 control panel is open, you can view and edit the rotoshape. You can press O over the Viewer to toggle the display overlay, if necessary. Click the right mouse button over any point to select options for the rotoshape.

Because this will be a garbage mask, we want to edit the shape to remove elements from the glass aquarium.

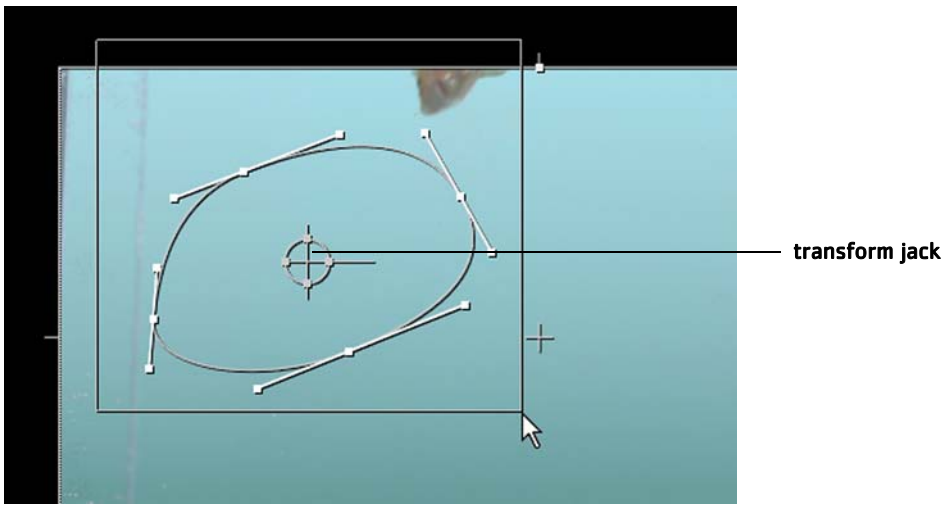
5. Drag the points and adjust the tangents—the handles on each of the points—to refine the rotoshape.

   Now we need to animate the garbage matte to follow the motion of the fish.

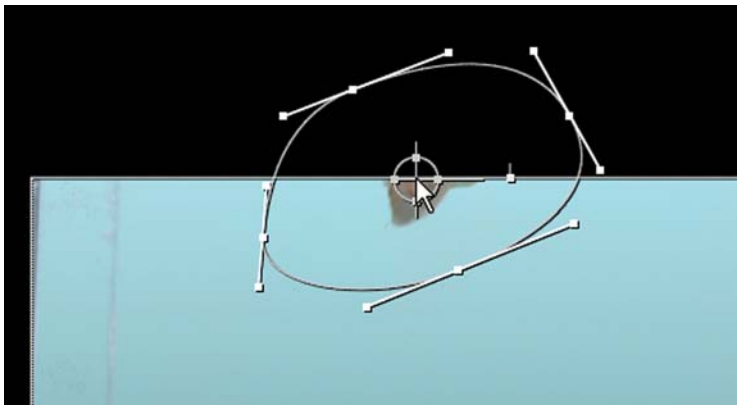
6. In the Bezier1 control panel, the autokey option should be active. If not, click the box for this option.

   The "shape 1 of 1" control shows the number of key frames—or distinct shapes—that are defined for the current bezier node along the timeline. Nuke will interpolate between these shapes as the clip plays.

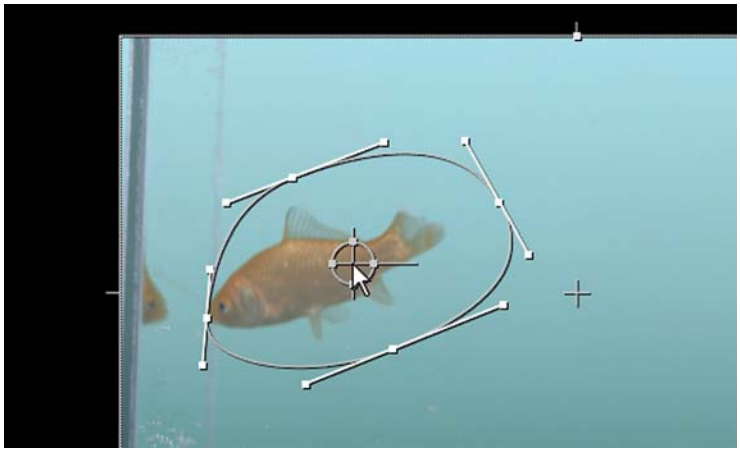
7. Move the time slider to frame **1** and drag a marquee around the entire bezier shape.



8. Drag the centre point of the transform jack, and move it over the current position of the goldfish.



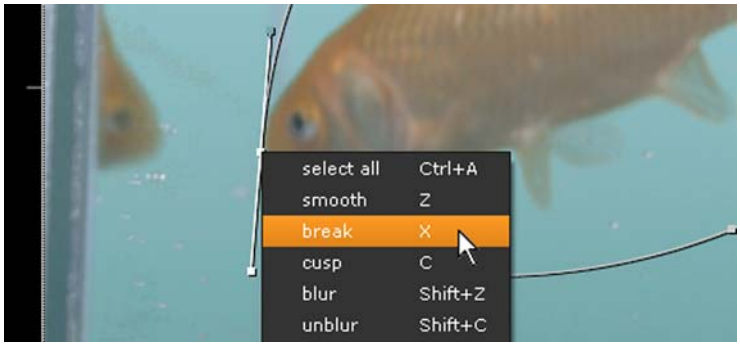
9. Go to end of the timeline, to frame **60**. Drag the shape once more to adjust for the movement of the goldfish.



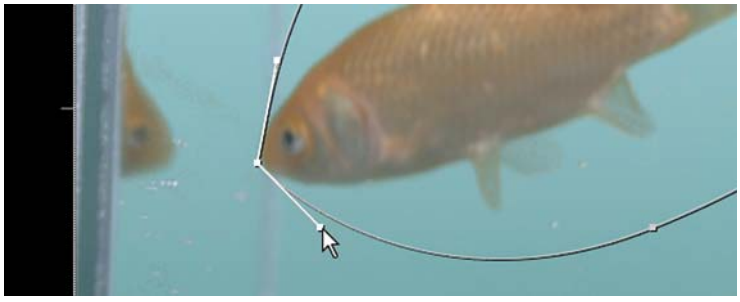
If your bezier shape is similar to the one shown above, then you probably don't need more than the three key frames at frames 1, 50, and 60.

However, you may want to scrub through the timeline and make adjustments.

10. Scrub to frame **60** on the timeline and you'll see the roto gets a little close to corner-line that we want to remove from the aquarium glass.
11. Click on an empty spot in the Viewer to deselect all points. Then, right-click on the point near the goldfish's nose and choose **break** from the pop-up menu, to break the point's tangent handle.



12. Adjust the handles to create a peak at the fish's nose.

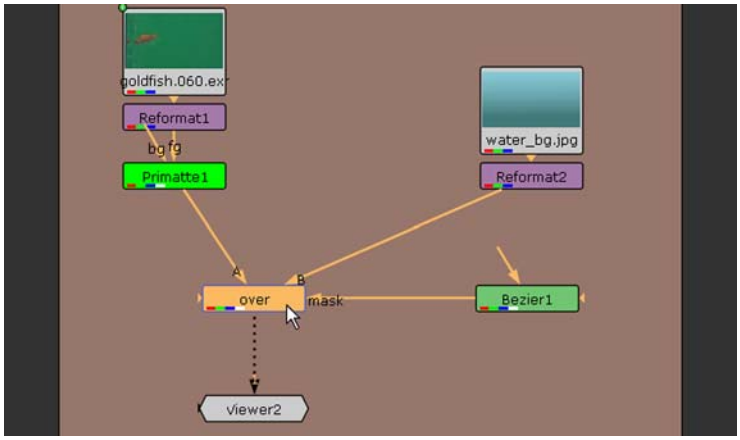


Now, for good measure, let's create a feathered edge for this particular point.

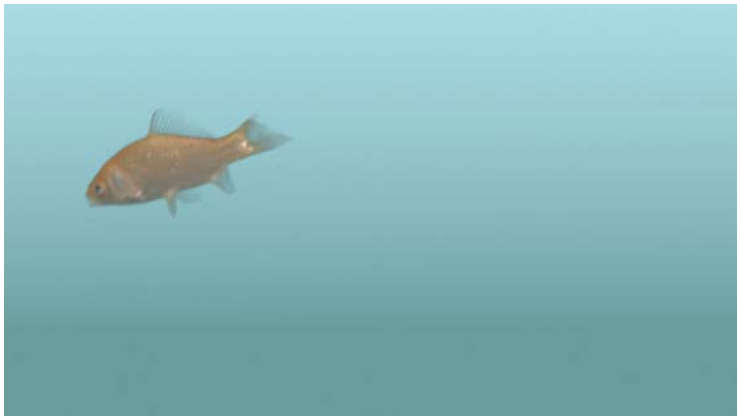
13. While pressing the **Ctrl** key (Mac user, press **Command**), drag the point away from the fish to create a feathered edge for this point, at this frame.

So you've drawn and animated the rotoshape. Let's wire it into the node tree to mask out the "garbage."

14. Drag the bg connector off the **Primatte1** node to disconnect it from the **Reformat2** node.
15. Choose **Merge > Merge** from the pop-up menu. Connect **Primatte1** to the **A** input on the over node. Connect **Reformat2** to the **B** input.



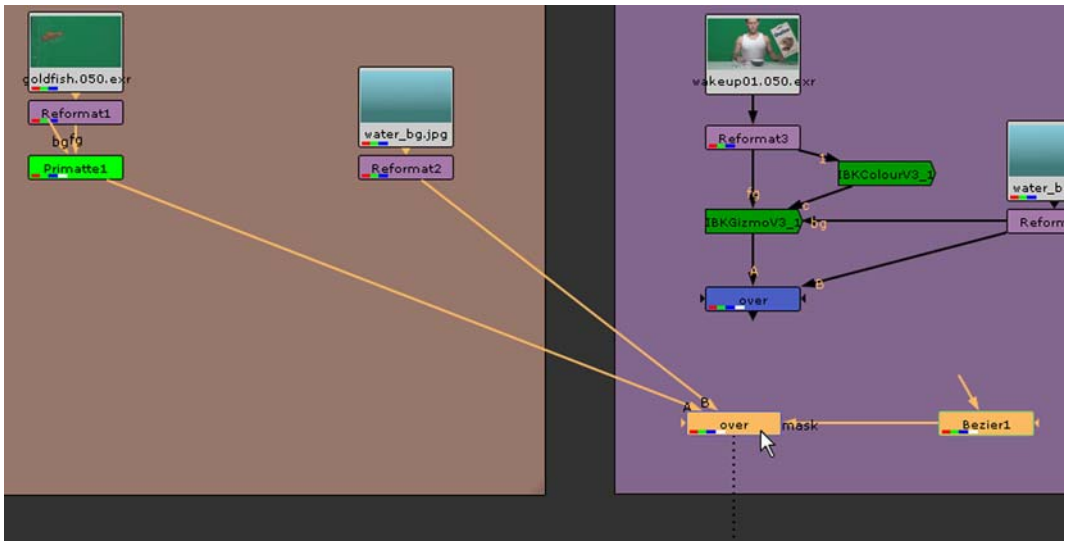
16. Connect the **Bezier1** node to the mask connector on the over node. This effectively removes the aquarium reflections in the image.



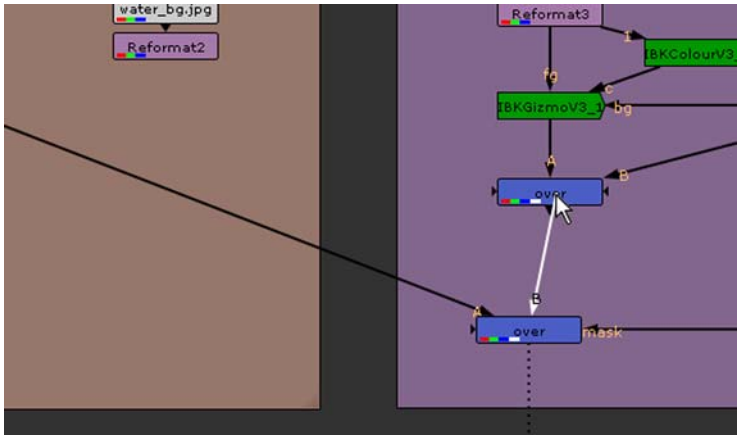
You might want to scrub through the timeline to see if there are places where you need to adjust the rotoshape.

If you want to take this a little further, you can now add the goldfish to the composite from the second example.

17. Select and drag the **over** node and the **Bezier1** node below the node tree you used for the IBK example.



18. Drag the **Viewer** node over, as well, and keep it connected to the **over** node.
19. Drag the **B** connector from the **Reformat2** node and connect it to the over node in the IBK node tree.



The Viewer shows the result. Of course you might want to add a **Transform** node after the first **over** node, to size and position the goldfish. Otherwise, this project is completed.

Keying Video

Nuke's Keyer node provides standard controls for pulling luma keys, green and blue screens, and colour channels. We'll use this keyer—and a few other nodes—to handle a special keying situation: video.

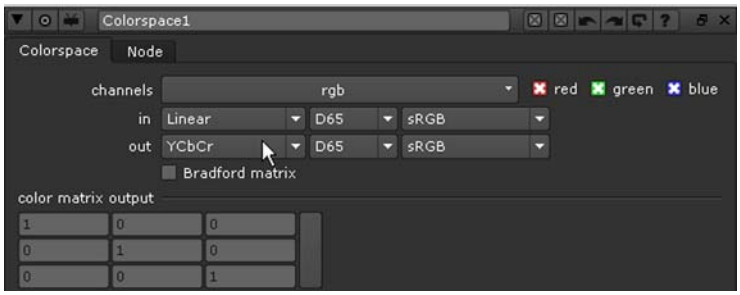
We'll begin by inserting a group of nodes that allow you to pull a cleaner matte by filtering the compression artifacts in the chroma red and chroma blue channels of digital video. This involves converting the image back to its original colourspace, blurring the channels with the artifacts, and then converting the image back to Nuke's native linear colourspace.

To prepare video footage for keying:

1. In the "**Keying Video**" node tree, select the **fgman.0001.dpx** node.

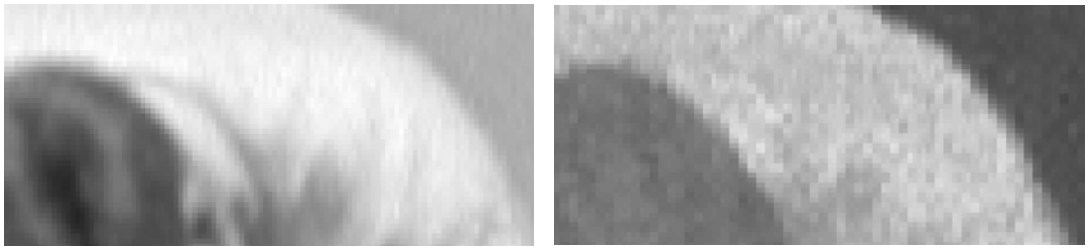
   When Nuke reads images into the workspace, it converts them to a linear colourspace. So here the first step is to convert the video image back to video YCbCr colourspace.

2. Right-click over the node tree and choose **Color > Colourspace** from the pop-up menu.
3. In the **Colourspace1** control panel, change the **out** parameter to **YCbCr**.



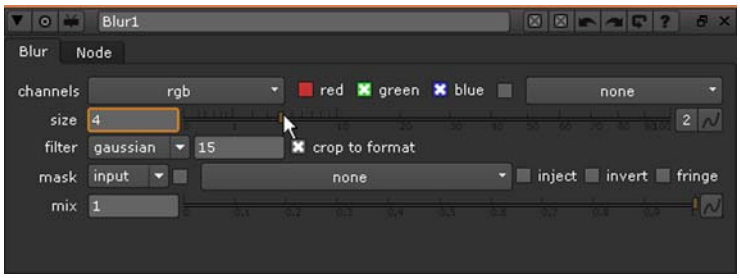
No, you're not having an 80's flashback. What you're experiencing is the result of the red, green, and blue channels remapped to the native video channels for luma (Y), chroma blue (Cb) and chroma red (Cr), respectively.

4. Press **r** over the Viewer to look at the Y channel. Press **g** to view Cb, and **b** to view Cr.



Above, you see the Y channel image on the left. The Cb channel image is shown on the right. Notice the "blocky" compression artifacts in the Cb channel. These make it difficult to get a clean edge for your matte, but since most of your detail is in the Y channel, you can apply a small blur operation to the Cb and Cr channels to improve the situation without losing much detail.

5. Press **r**, **g**, or **b** again to toggle back to all colour channels.
6. Right-click on the Colourspace1 node and choose **Filter > Blur**. In the Blur1 control panel, set the blur **size** to **4**.

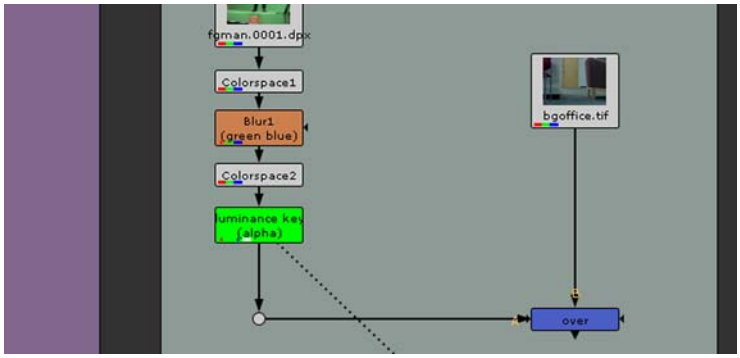


7. Select **rgb** in the **channels** menu and uncheck the **red** channel box. You don't want the blur operation to process the image in the red channel (the remapped Y or luma channel) because this channel is uncompressed.
8. Right-click on the Blur1 node and add another **Colour > Colourspace** node. Change the **in** parameter to **YCbCr** and the **out** parameter to **Linear**. This converts the image back to standard rgb/linear.

To pull a basic greenscreen key:

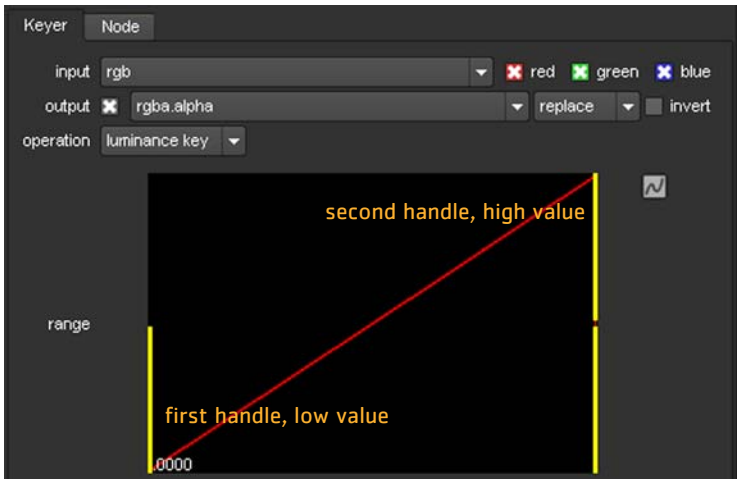
1. Select the **Colourspace2** node, and choose **Keyer > Keyer** from the right-click menu.

   This inserts a keyer named "luminance key," which is the default keying operation for this node.



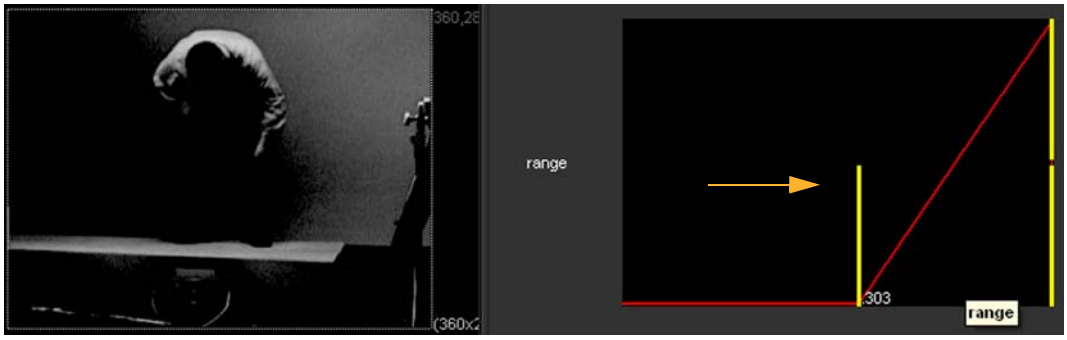
2. Attach a Viewer to the **luminance key** node and then press **A** to display the alpha channel.

   In the control panel for the Keyer node, you'll see the "range" graph:



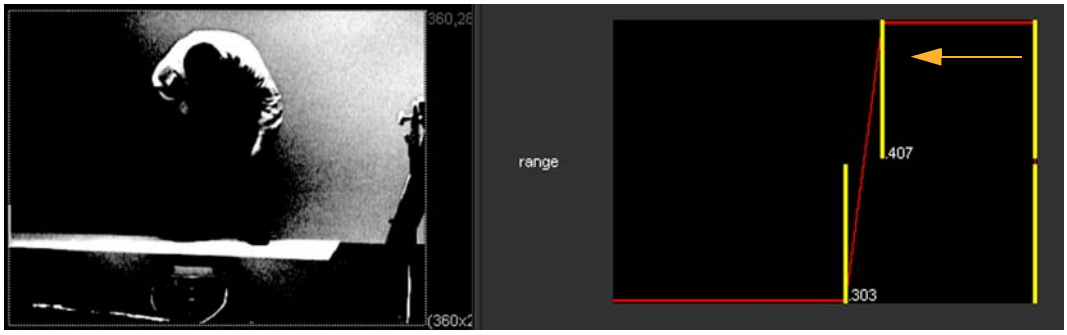
   The range graph is where you'll adjust the low and high pixel values of the matte. The first yellow handle on the left determines the low or transparent values of the key and second handle, on the upper-right, determines your high or opaque values.

3. Drag the first yellow handle to the right until it reads **.303** (approximately), and watch the effect in the Viewer.



This sets the low value for the matte. Any pixels that fall below this value are clipped to black.

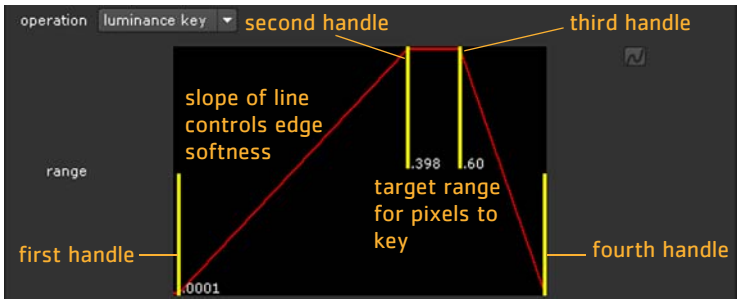
4. Drag the yellow handle, located at the upper-right edge, to the left until it reads **.455** (approximately).



This sets the high value for the matte. Pixel values above this setting are clipped to white. At these settings, it's not quite "matte-worthy," so let's make an adjustment.

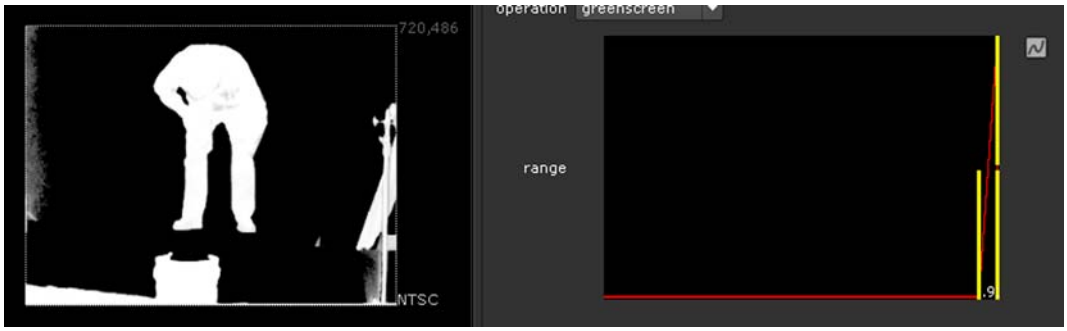
5. Drag the first handle to change the low setting from **.303** to **.424**, and drag the second handle to change the high value from **.455** to **.61**.

As you adjust the location of the handles, the slope of the line controls the softness, or level of greys, for the matte edge. A gradual slope produces a softer edge. A sharper slope produces a jagged or crunchy edge—drag the first handle on top of the second handle and you'll see what that means.



The default positions let you control the low and high values, assuming your image has distinct light and dark areas. However, sometimes the subject of the matte falls into the "middle-grey" area; the third and fourth handles on the curve, after the first two, let you shift the centre for the high values of the key.

6. Change the keying **operation** to **greenscreen**. Pull the high value handle to the right, as far at it will go (setting = 1.0). Then, set the low value at **.90**.

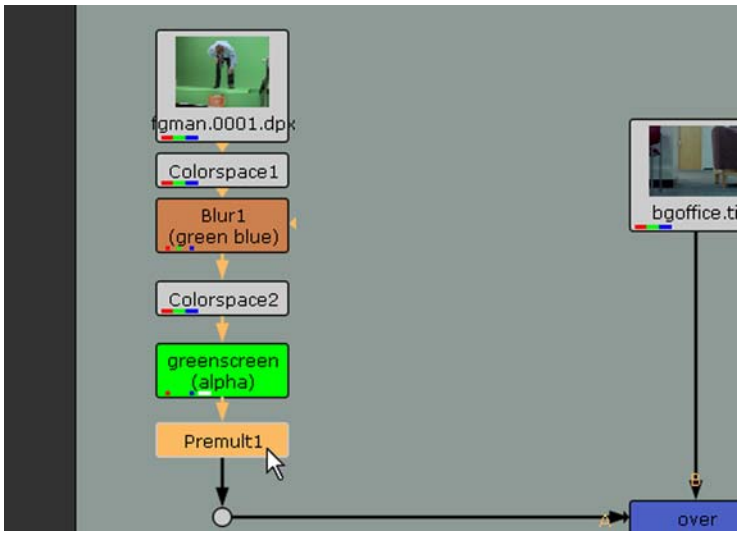


There's a lot of garbage around the image, but it looks like you've got a fairly clean edge around your subject. Let's check it in the comp.

7. Switch the Viewer to display all colour channels and then attach the Viewer to the **over** node.

That's completely terrible. What happened? The alpha channel created by the greenscreen keyer must be multiplied into the pixel values of the original image in order to generate the matte. Some keyers, like Primatte, provide a "composite" output option which handles the multiplication for you. The Keyer node does not, so you'll have to do it manually.

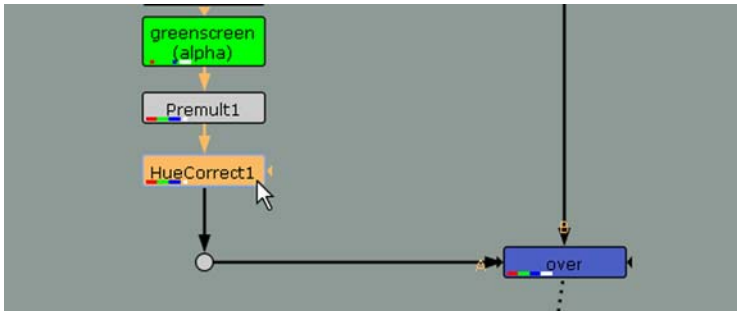
8. Select the **greenscreen** keyer node and choose **Merge > Premult** from the right-click menu.



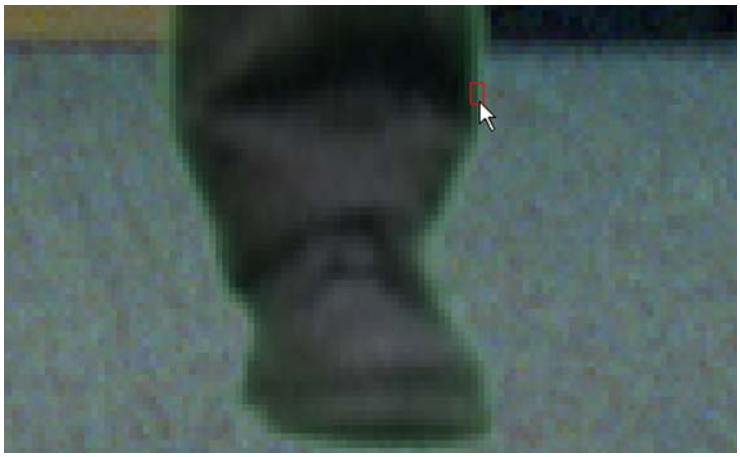
9. Adjust the high and low ranges values in the greenscreen node's control panel to refine the edges around your subject.

This keyer isn't particularly good at handling spill, so let's add a colour-correction node to remove those green edges.

10. Select the **Premult1** node and then choose **Colour > HueCorrect** from the right-click menu.

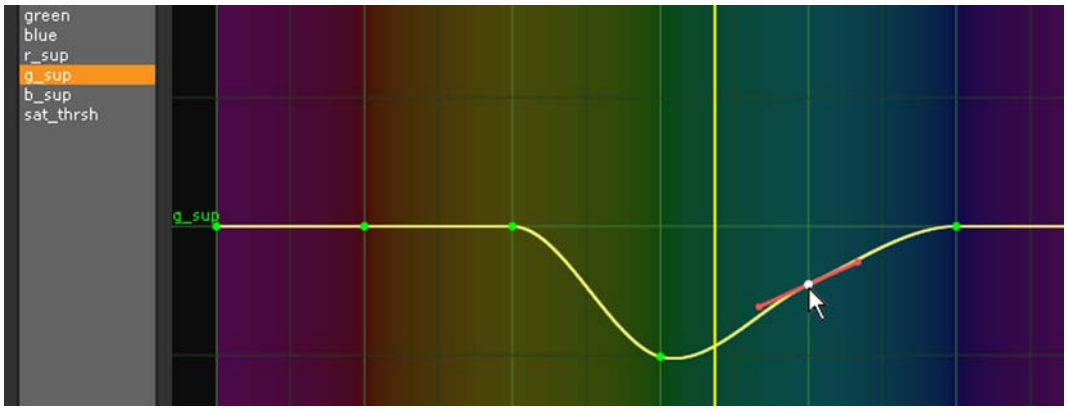


11. Select the **g_sup** parameter in the **HueCorrect1** control panel to select the green suppression curve.
12. Over the Viewer, press the **Ctrl** or **Command** key and scrub over the green edges.



Inside the HueCorrect1 control panel, you'll see a yellow vertical line, which marks the place in the curve that you need to lower to suppress the samples pixels.

13. While viewing the edges you want to suppress, adjust the **g_sup** curve so that it looks similar to this:

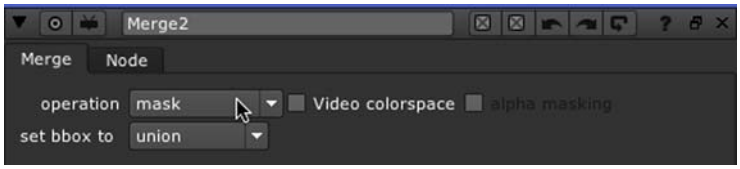


When you're satisfied with the spill-suppression, you may want to add a quick garbage matte to remove the rigging.

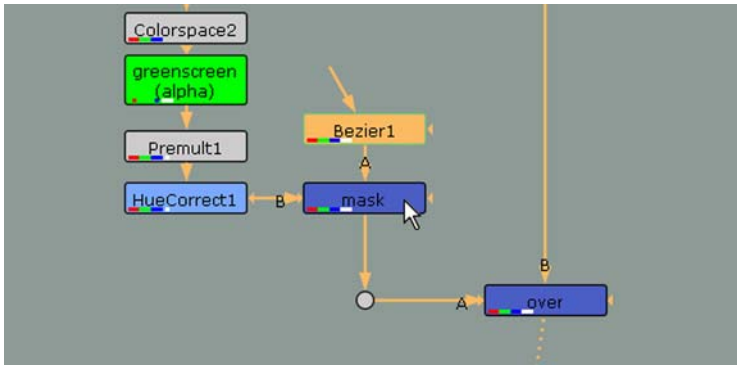
14. Click on an empty place in the node tree, and add a **Draw > Bezier** node.

15. While pressing the **Ctrl+Alt** keys (Mac users press **Option+Command**), click over the Viewer to draw a bezier shape around the man. Four points should be enough.

16. Select the Bezier node and press **M** to add a Merge node— that's the shortcut for choosing **Merge** > **Merge** from the menu—and change the merge **operation** to **mask**.



17. Rewire the nodes as shown below. The **Bezier1** node should be connected to the A input on the mask node, **HueCorrect1** should be connected to the B input. Connect the output of **mask** to the "elbow" dot.



This method of masking is a little different than what you did in the previous example. The point here is that there are different ways to structure these types of composites. Your results should look similar to the screen capture below.

Epilogue

Keying is rarely a simple matter of picking the screen colour you want to remove. To get the very best mattes, you often need to combine several techniques and you've learned several in this chapter. You've pulled mattes with Primatte and Nuke's Image-based Keyer, and you've used the rotoscoping tools to cleanup a matte and control the parts of the image you want to use in the composite.

You've also seen how to key video footage by converting the image to its native colourspace and filtering the compressed channel data to pull a cleaner matte.

- End of Tutorial -

Tutorial 4: 3D Integration

Nuke's 3D workspace creates a powerful compositing environment within your project script. This workspace combines the advantages of cameras, lighting, and a three-axis (x, y, and z) environment, with the speed of node-based compositing. You pipe 2D images into the 3D space, setup a camera, animate your scene, and then render the results back to the 2D composite.

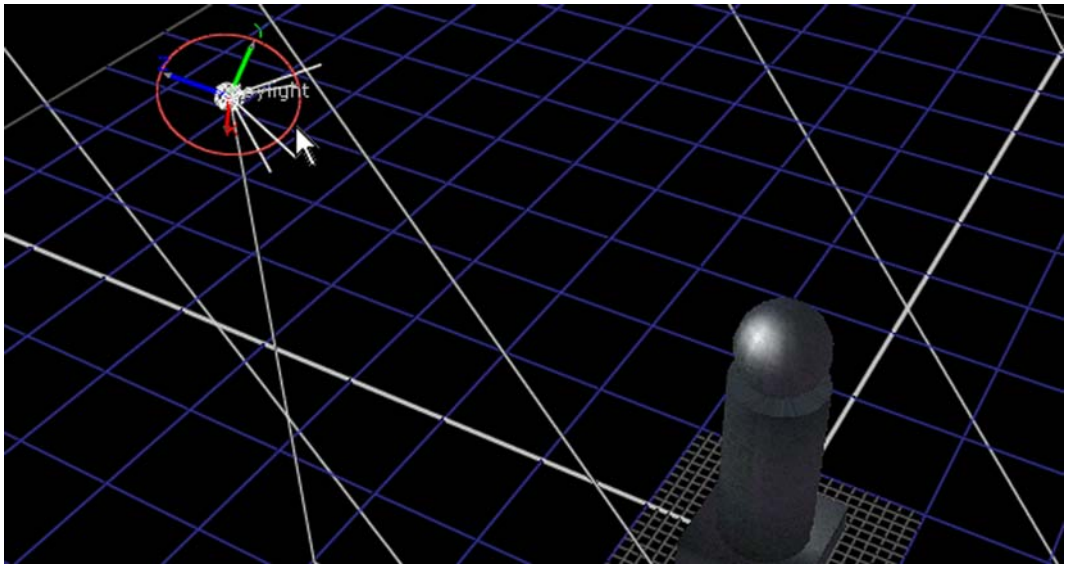


Figure 4-1: Compositing in the 3D workspace

The Basic 3D System

The 3D workspace is defined by a group of nodes in your script. The most basic setup includes a Camera node, a Render node, a Scene or Geometry node, and nodes that provide the 2D images you want to pipe into the 3D compositing space.

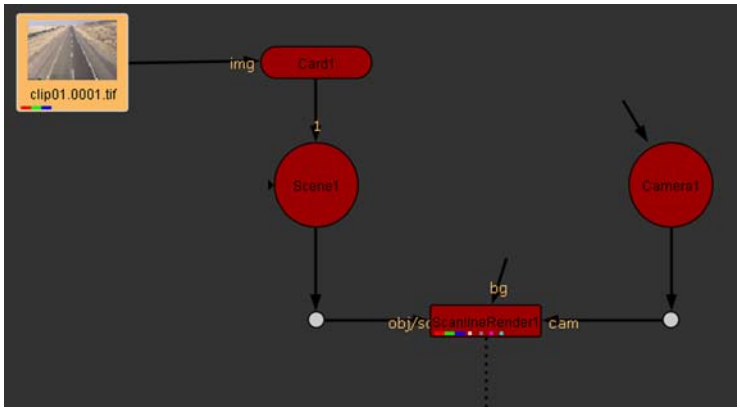


Figure 4-2: The basic 3D node tree: 2D image, geometry, scene, render, and camera

The 3D Viewer

Once you have the 3D node structure, you can use any Viewer in Nuke as a gateway to the 3D compositing space. Choose **3D** from the view list, or press the **Tab** key over the Viewer to tog-

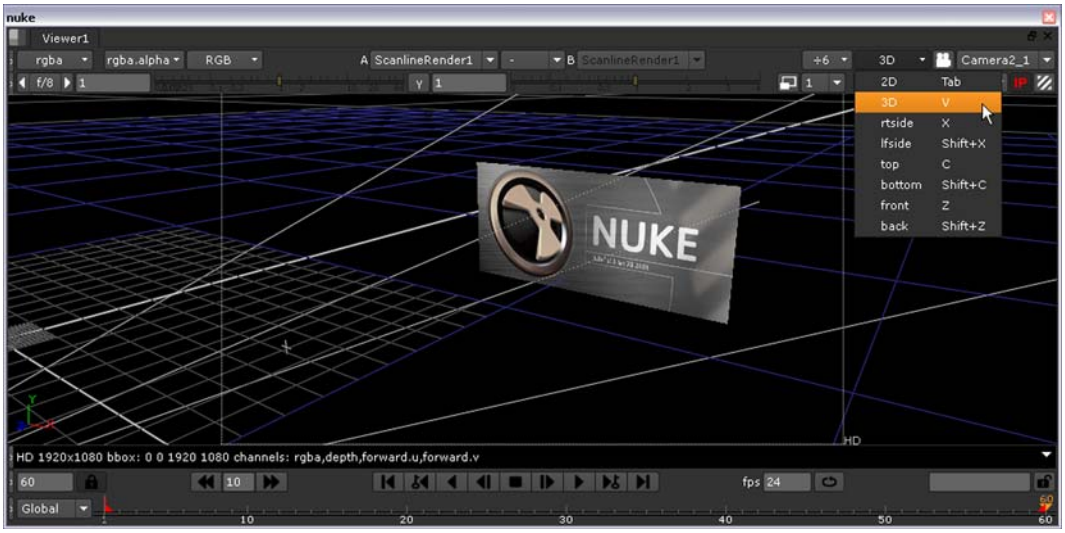


Figure 4-3: Each Viewer window can also display the 3D workspace

On the view list, you'll also see orthographic views—rtside, lfside, top, bottom, front, back—which provide non-perspective views into the scene. In three-quarter perspective, it can be difficult to accurately place objects on the axes, and the non-perspective views make it easier to line things up.

The Geometry or Scene Node

Every 3D system needs a piece of geometry—a card, a sphere, a cube, something— to receive an image or clip that the camera can "see." One is all you need, but you can setup complex systems with a large amount of 3D data. When you have two or more objects for a 3D system, you need a Scene node to create a "place" where the camera (and the ScanlineRender node) can see all the objects at once.

The Camera Node

The Camera node creates your view into a scene. It has several controls to help you match the properties of a physical camera. You can animate its position or import animation or tracking data to matchmove your 3D scene with a background plate. A 3D system can have multiple cameras connected to the Scene node, to create different views on a 3D scene.

Tip

Only one camera can be connected to the ScanlineRender node to generate the output, but you can insert multiple ScanlineRender/Camera node pairs to generate output from various perspectives.

The ScanlineRender Node

The last node, ScanlineRender, sends the results of your 3D scene back into your composite as a 2D image. It's always 2D in, 3D manipulation, and then 2D back out, which is why this is often called "2-and-a-half-D."

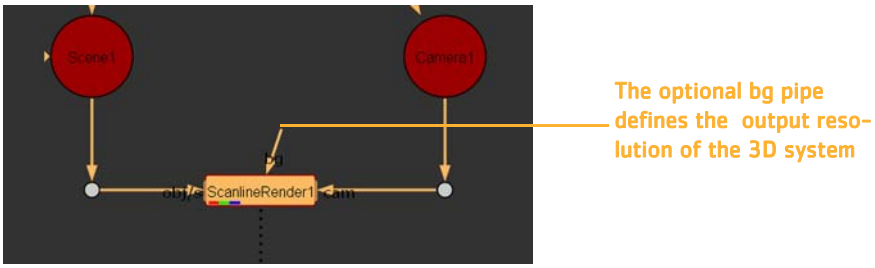


Figure 4-4: The scanline render node converts the image back to 2D

The image created by the ScanlineRender node will be the same resolution as your project settings. When you need to render a specific resolution, use the optional "bg" pipe. Connect a Constant node with the resolution you want and that will define the output of the ScanlineRender node.

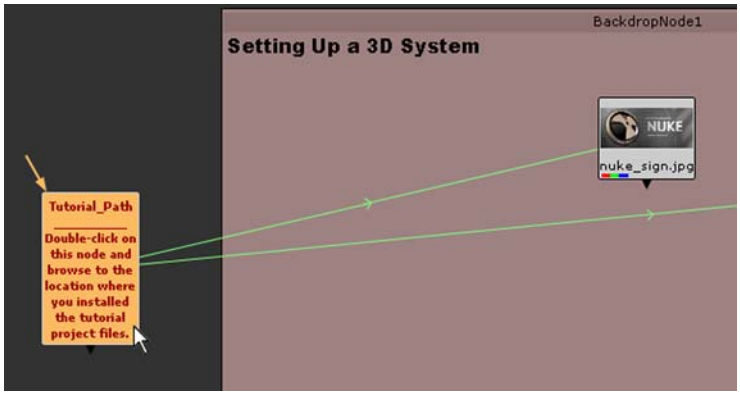
So now you know the basic 3D setup for your compositing script. Let's take a test drive.

Open the Tutorial Project File

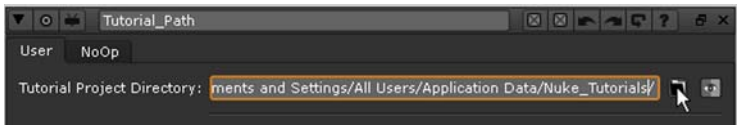
The "3Dinteg_tutor.nk" project file includes the node trees for the first part of this chapter.

To open the project file:

1. Launch the Nuke application and choose **File > Open** from the menu bar.
2. In the file browser, navigate to your **Nuke_Tutorials/3DInteg/** folder, select the **3dinteg_tutor.nk** project file and click **Open**.
3. Locate the **Tutorial_Path** node, on the left side of the script, and double-click it to open its control panel.


BackdropNode1
Setting Up a 3D System
NUKE
nuke_sign.jpg
Tutorial_Path
Double-click on this node and browse to the location where you installed the tutorial project files.

4. Click the "file folder" button. Browse to the location where you installed the tutorial project files, and then click **Open** to select the location.



After you select the correct path, the error messages should clear from the Read nodes, and the thumbnails in the script will update with the correct images.

5. Close the **Tutorial_Path** control panel. Then, choose **File > Save As** to save a copy of the project file.
6. Move the mouse pointer over the node graph, and press **F** to frame the entire contents of the project file.

   The green arrows (lines) show the links between the Tutorial_Path node and the Read nodes.
7. If you wish, press **Alt+E** to hide the expression arrows.

The Tutorial_Path node saves the location of the project files on your computer, so you don't need to repeat this for future sessions with this project file.

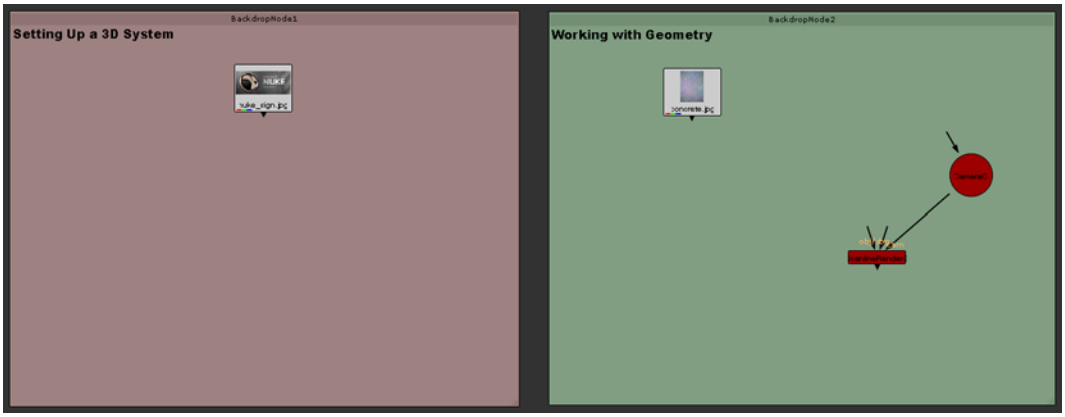


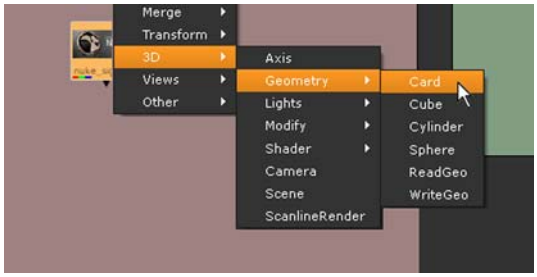
Figure 4-5: Node trees in the 3dinteg_tutor.nk project file

Setting Up a 3D System

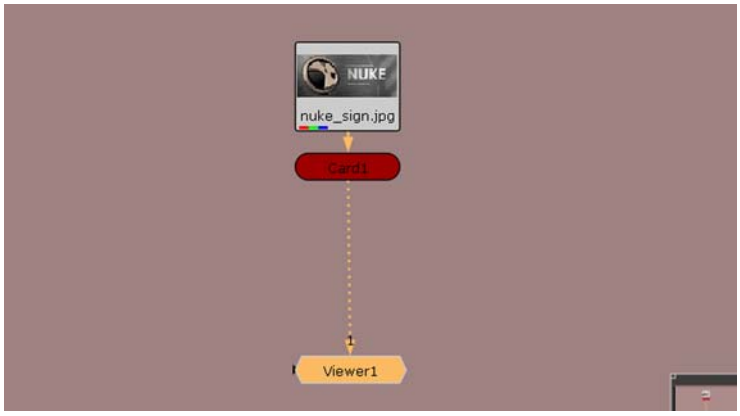
Let's start with the basics. In this first example, you'll create a basic 3D node tree, map an image to a 3D card, manipulate it, and then render the result back out to the 2D composite.

To setup a 3D node tree:

1. In the "3Dinteg_tutor.nk" project file, locate the backdrop node labeled "**Setting Up a 3D System**." You'll see a Read node with the image you'll use for this example.
2. Right-click over the **nuke_sign.jpg** node, and choose **3D > Geometry > Card**.

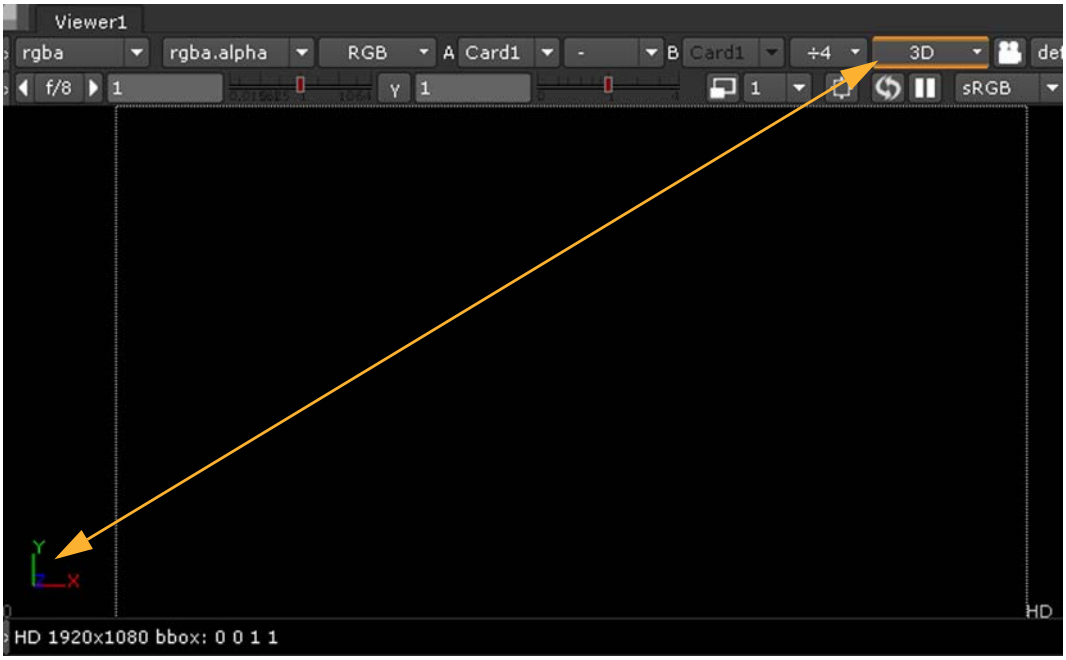


This attaches a "Card1" node. Let's see what it looks like in 3D.



3. Attach a Viewer to the **Card1** node and Nuke switches the Viewer to 3D.

Wow, that's amazing. It looks exactly like the 2D Viewer. How can anyone tell the difference? Check the lower-left corner of the Viewer and you'll see an orientation marker for the three axes in 3D. You'll also see that "3D" is displayed on the view list.

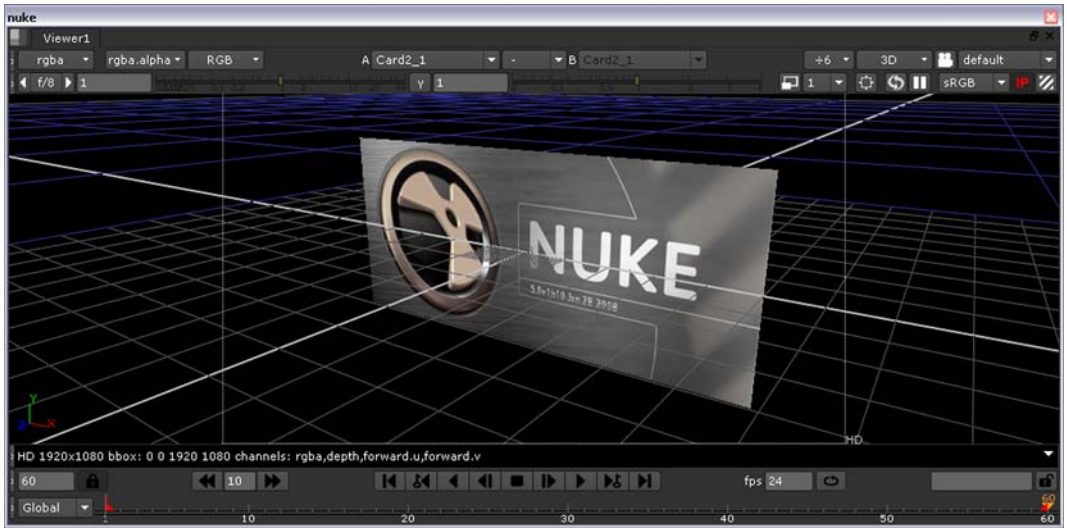


That sign is a little darker than expected, isn't it? Actually, you can't see the image yet because the default view of the 3D workspace is at the origin or centre of the space. Perhaps zooming-out will improve the view.

4. Press the **Alt** key (Windows /Linux) or the **Option** key (OS X), and drag with the middle mouse button to zoom or "dolly." Drag to the left and you'll zoom out.

Hey, look. There's the Nuke emblem. In the 3D Viewer, the "pan" and "zoom" controls are exactly the same as what you've used for the node tree and the 2D Viewer, but let's try "tumbling" to get a better view.

5. **Alt-** or **Option**-drag with the right mouse button to rotate around the origin point of the 3D workspace. You now see the 3D grid and the image mapped to the card.



When an image is connected directly to a Card node like this, it is applied as a flat or "planar" map. The size of the card adjusts to the dimensions of the image.

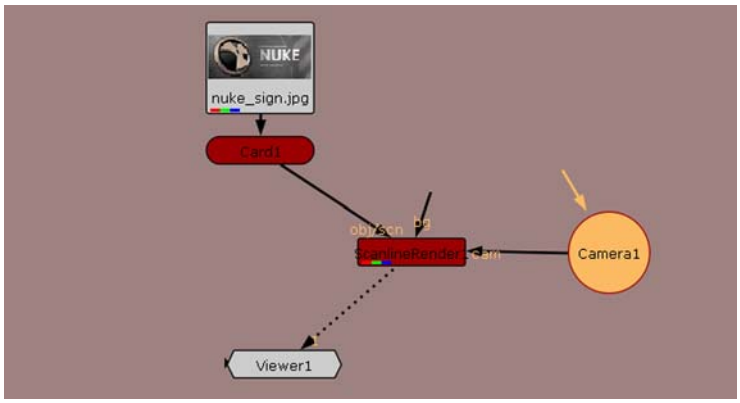
6. Click on the card and you will select the node in the node tree and also the card inside the 3D workspace.

7. Use the mouse (and the **Alt** or **Option** key) to navigate through the workspace. Go ahead, pan, dolly, and rotate at will. Then, press **F** over the Viewer to frame the 3D view.

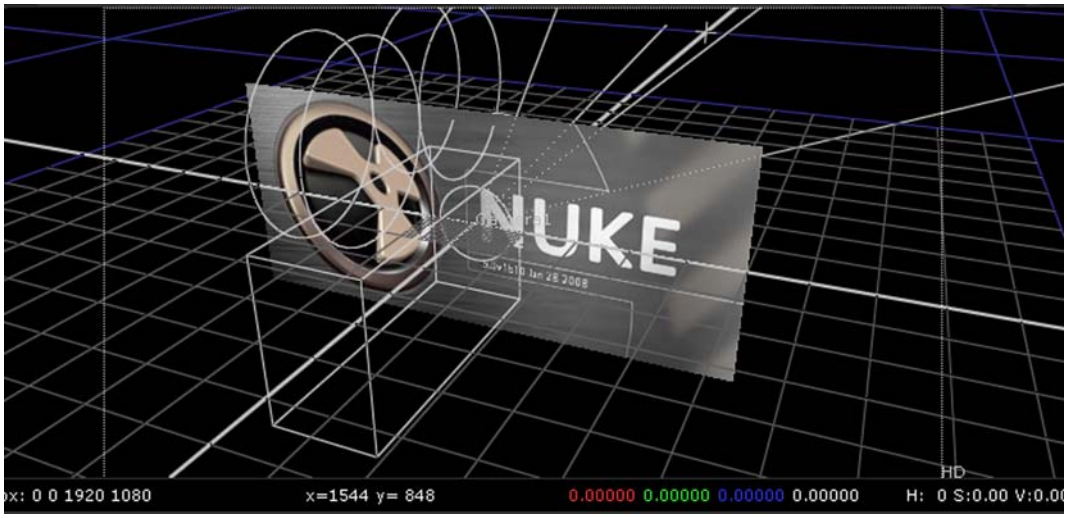
Tip

If you don't like the standard navigation controls, open the **Preferences** control panel (**Shift+S**), select the **Viewers** tab and change the **3D control type** to **Maya**, **Lightwave**, or **Houdini**.

8. Click on an empty spot in the node graph to deselect all nodes. Let's add the other nodes you need.
9. Right-click on the node graph and choose **3D > Camera** from the pop-up menu. Keep its control panel open so you can manipulate the camera in the Viewer.
10. Right-click and choose **3D > ScanlineRender** to insert a render node, and then connect the nodes as shown below.

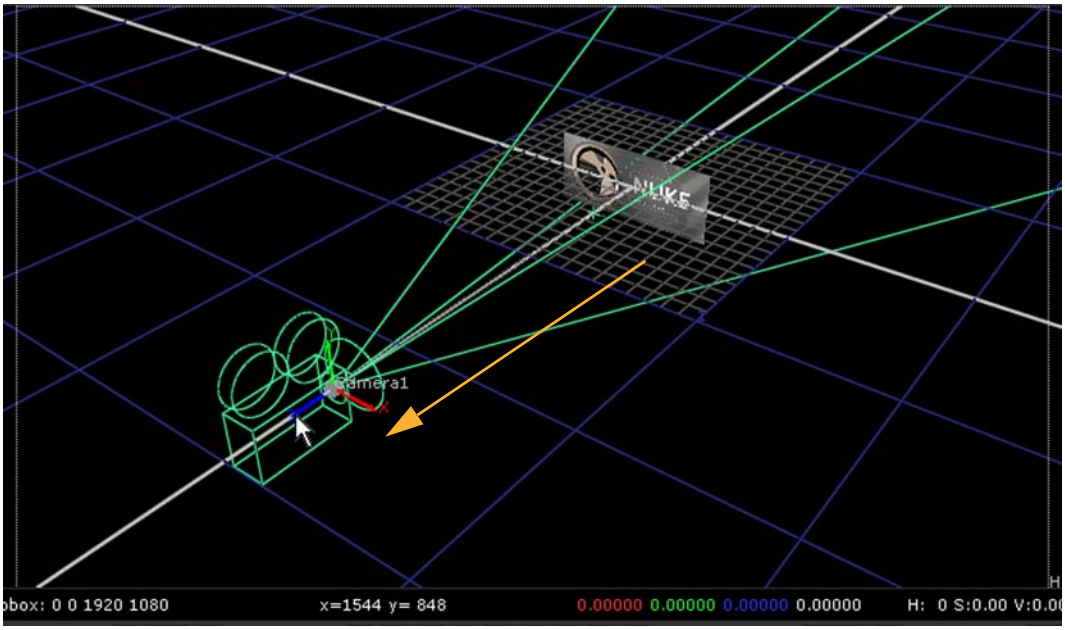


11. Connect the Viewer to the **ScanlineRender** node, and you have the most basic 3D system in Nuke.
12. Press **Tab** over the Viewer to change to the 2D view. You won't see the Nuke emblem—hey, where did it go? We saw it before.
13. Press **Tab** again to switch back to 3D. You'll see the default camera position is too close to view the card. Let's move things around to get an image for 2D.

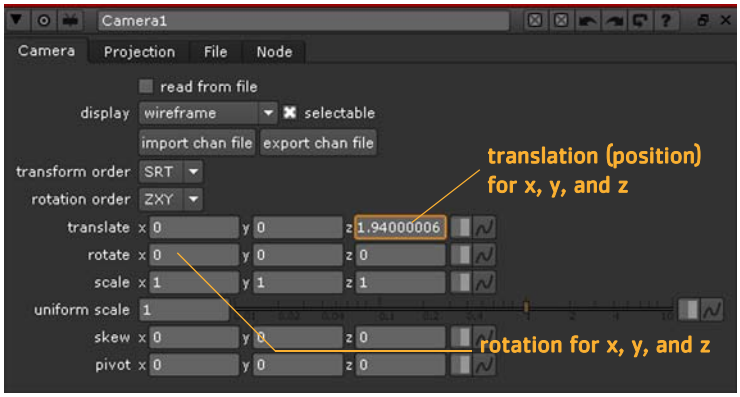


To position objects in the scene:

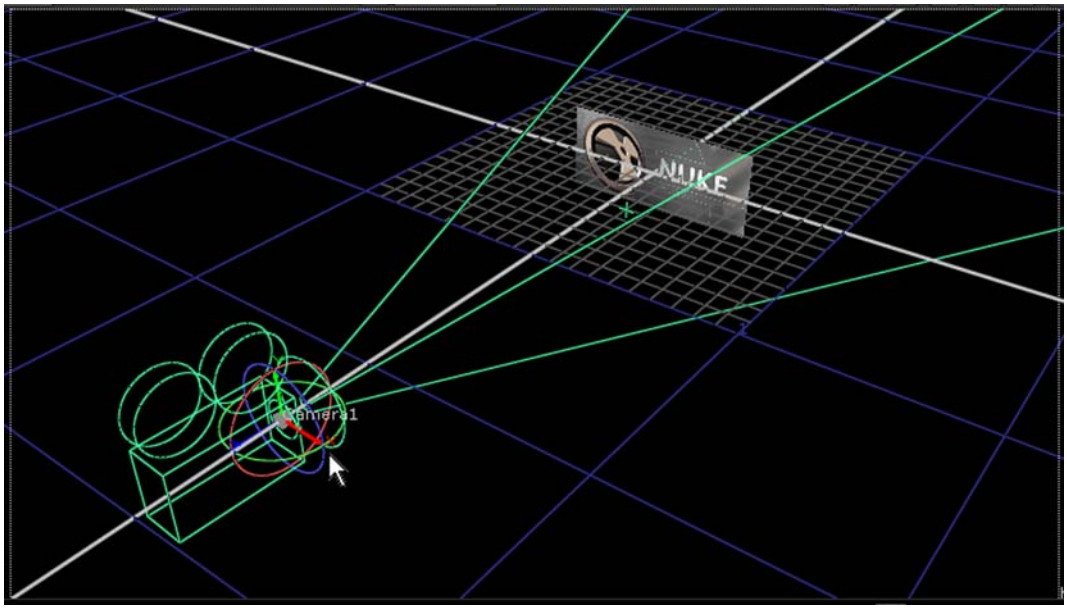
1. **Alt-** or **Option-**drag with the middle mouse button to dolly out and show more of the 3D workspace.
2. Select the camera. You can do this by clicking the camera object in the Viewer or clicking the **Camera1** node in the node graph.
3. Drag the transform handles to move the camera away from the card, along the z-axis.



As you drag the camera, look at the camera's control panel. You'll see the x/y/z transform values reflect the camera's current position.



4. Press the **Ctrl** key (Mac users press **Command**) over the Viewer and the transform handles change to rotation rings.



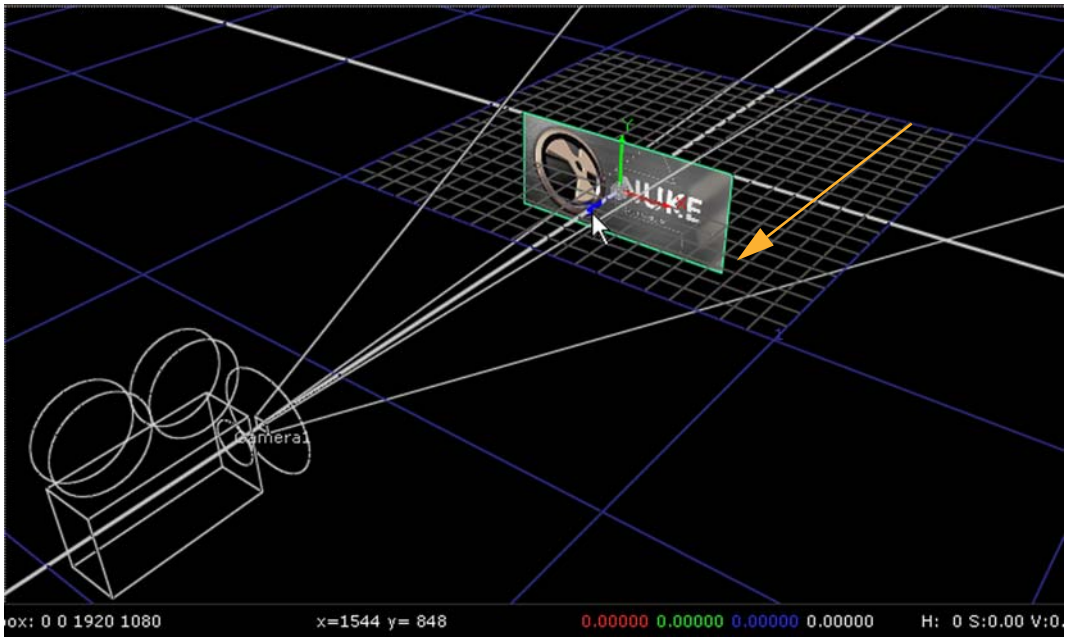
5. Drag the green ring to rotate the camera around the Y-axis. Notice the x/y/z rotation values in the control panel reflect the angle of the rotation.

   The blue handle "rolls" or rotates on the Z-axis, and the red handle rotates on X.

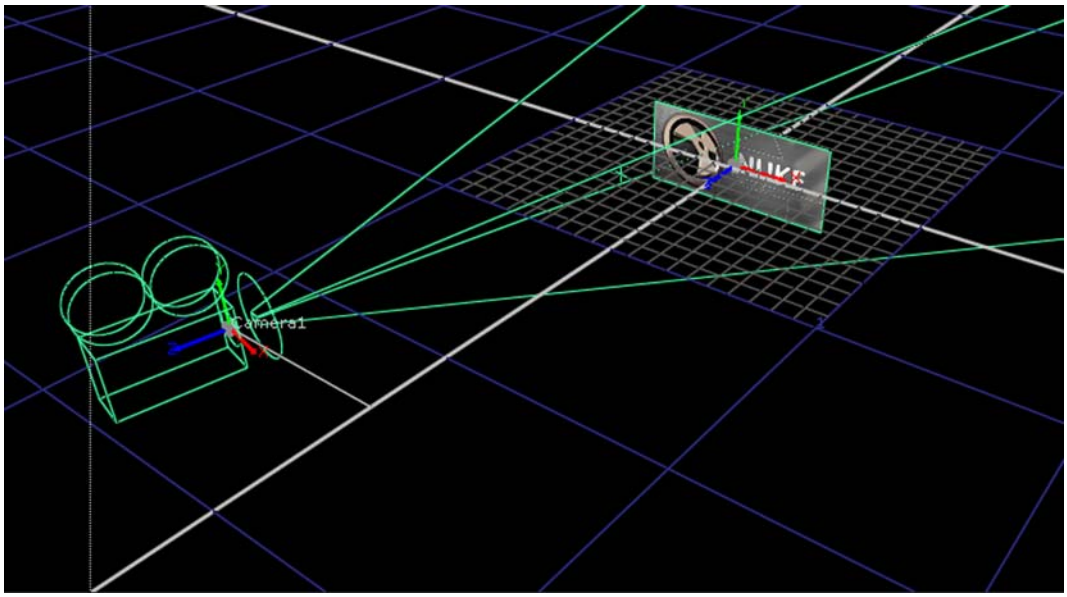
6. Now, select the card object and move it away from the camera.

   Keep the control panel open for the Card node. As with the Camera node, the transform handles disappear from the Viewer when you close the control panel.

7. Drag the card's transform handles to position it in the 3D workspace. If you wish, press the **Ctrl** key (Mac users press **Command**) over the Viewer and rotate the card.



8. Press **Tab** over the Viewer to switch between the 2D and 3D views to see the image the ScanlineRender node will produce.



9. Before you continue to the next example, close all the control panels that are currently open.

   In this example, it doesn't matter where you move the camera or card. In reality, however, you often need to use specific values, which you can enter directly in the control panels.

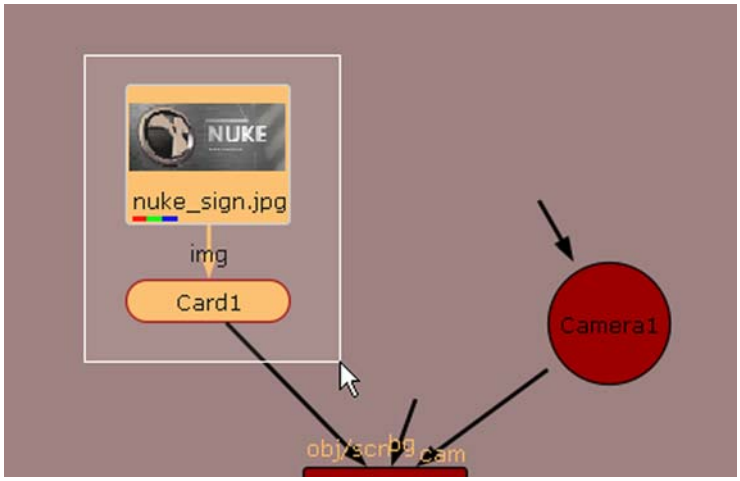
   You can also import camera data or animation curves—did you notice the **import chan file** button in the camera's control panel?—and apply them to the objects in the workspace.

Making a Scene

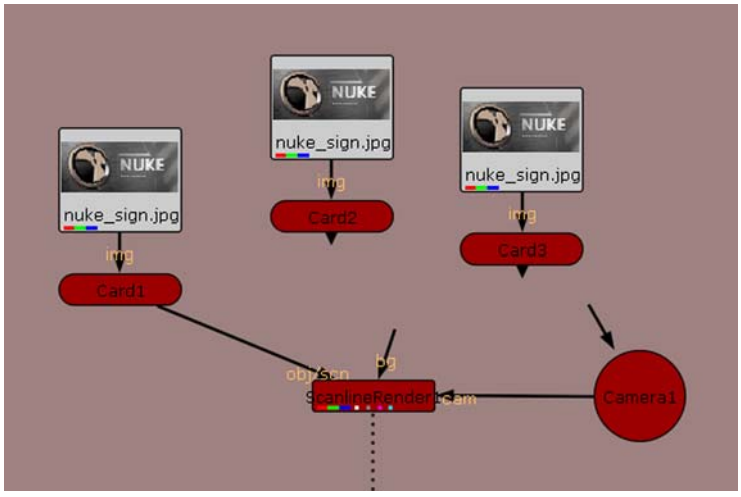
We mentioned earlier the Scene node creates a place where multiple objects may be seen by the camera and the render node. If you only have a single object, you don't need a Scene node, but where's the fun in that? Scene nodes make it possible to really tap into Nuke's ability to handle huge amounts of 3D information, and you should know how to use it.

To setup a scene:

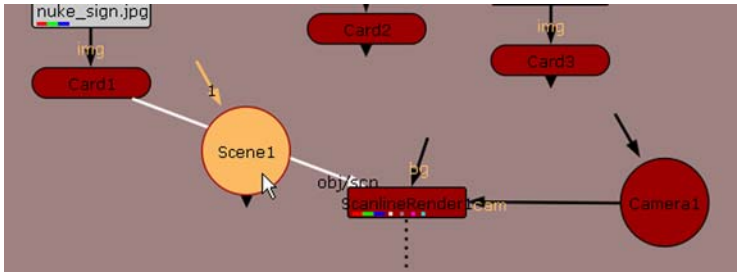
1. Inside the "Setting Up a 3D System" node tree, drag a selection around the **nuke_sign.jpg** node and the **Card1** node to select them.



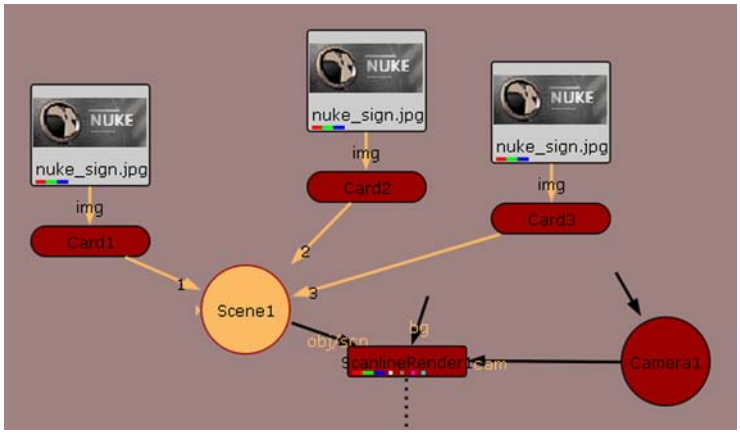
2. Press **Ctrl+C** (Mac users press **Command+C**) to copy the selected nodes or choose **Edit > Copy** from the pop-up menu.
3. Press **Ctrl+V** (Mac users press **Command+V**) or choose **Edit > Paste** to insert a copy of the nodes. Press **Ctrl+V** or **Command+V** again to insert a second copy, and then arrange the nodes as shown below.



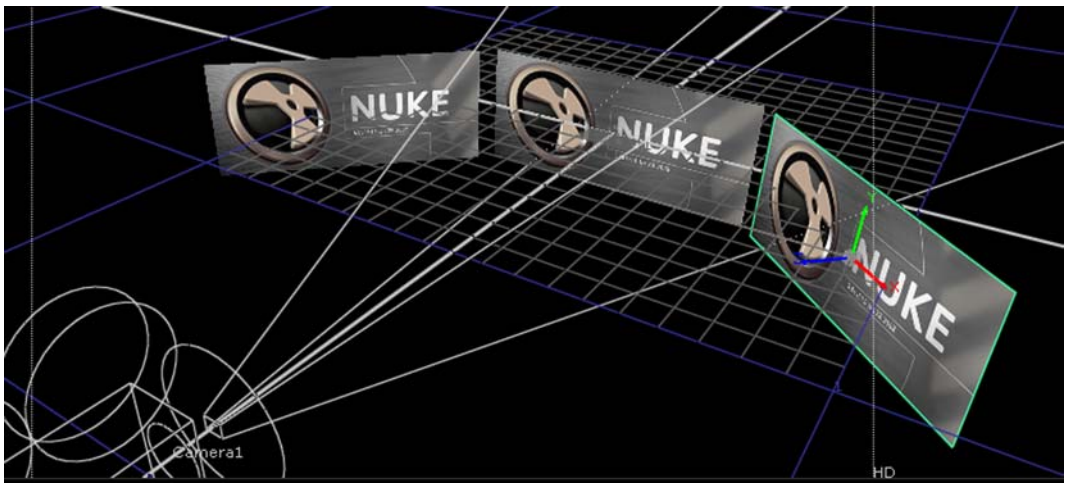
4. There are multiple cards now, and you need a Scene node to create a space where the rendering node can see all cards at once.
5. Click on an empty space in the node graph to deselect all nodes. Right-click and choose **3D > Scene** to insert the "Scene1" node.
6. Drag the **Scene1** node onto the obj/scn connector to insert the Scene1 node between **Card1** and **ScanlineRender1**.



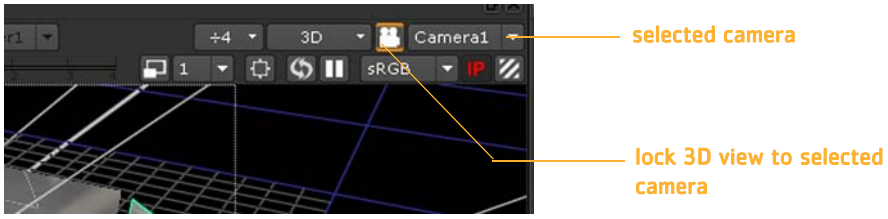
7. Connect the **obj/scn** connector from the **ScanlineRender1** node to the **Scene1** node. Connect each **Card** node to the **Scene1** node.



8. Double-click on the **Card1** node to open its control panel. In the Viewer, you'll see the transform handles for the first card.
9. Move and rotate the card to a different position. For this example, it doesn't matter where you place it.
10. Open the control panel for **Card2** and move its card to a different place in the scene.
11. Open the **Card3** control panel and move that card, also.



You could switch to the 2D view to see the result, but why not just look through the camera? Next to the view list, you see the button that locks the 3D view to the camera.



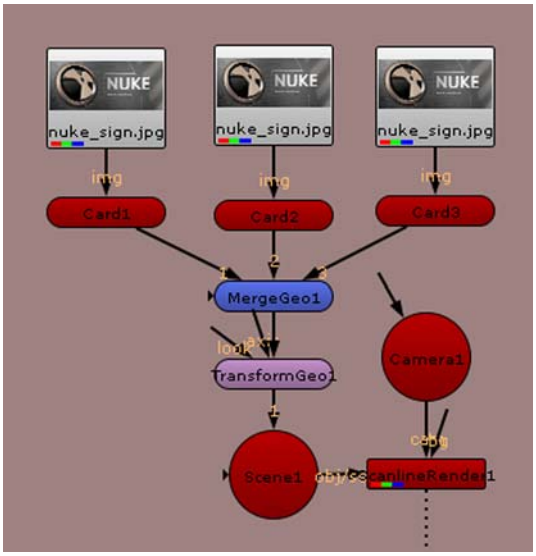
12. From the view list, choose **3D** (**V**) to switch to a 3D perspective view. Then click the "lock view to 3D camera" button.

13. Turn off the "lock 3D view to camera" button. You won't need it during the rest of this tutorial.

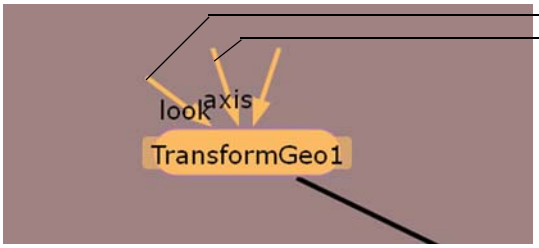
Merging and Constraining Objects

You can merge objects and move them together as a group. To do so, you need to insert MergeGeo and TransformGeo nodes after the objects. The MergeGeo node first merges the objects together, after which you can use the controls of the TransformGeo node to move the merged objects in the 3D space. You can also use the TransformGeo node to constrain objects, as you will notice later in this tutorial.

To merge the three card objects together, right-click on the Card1 node and select **3D** > **Modify** > **MergeGeo**. This inserts a MergeGeo node between Card1 and Scene1. Disconnect the Card2 and Card3 nodes from the Scene node and connect them into the MergeGeo node. Then, right-click on the MergeGeo node and select **3D** > **Modify** > **TransformGeo**. Your node tree should now look like the following:



On the TransformGeo nodes, you see multiple connectors. The connector without a label should be attached to a geometry object or a MergeGeo node. The other connectors act as constraints on the connected object's position.


look
axis
TransformGeo1
You can connect "axis" to an Axis node. This links the position of the current object to the position of the Axis node.
You can connect "look" to the object or camera that the current object should "face."

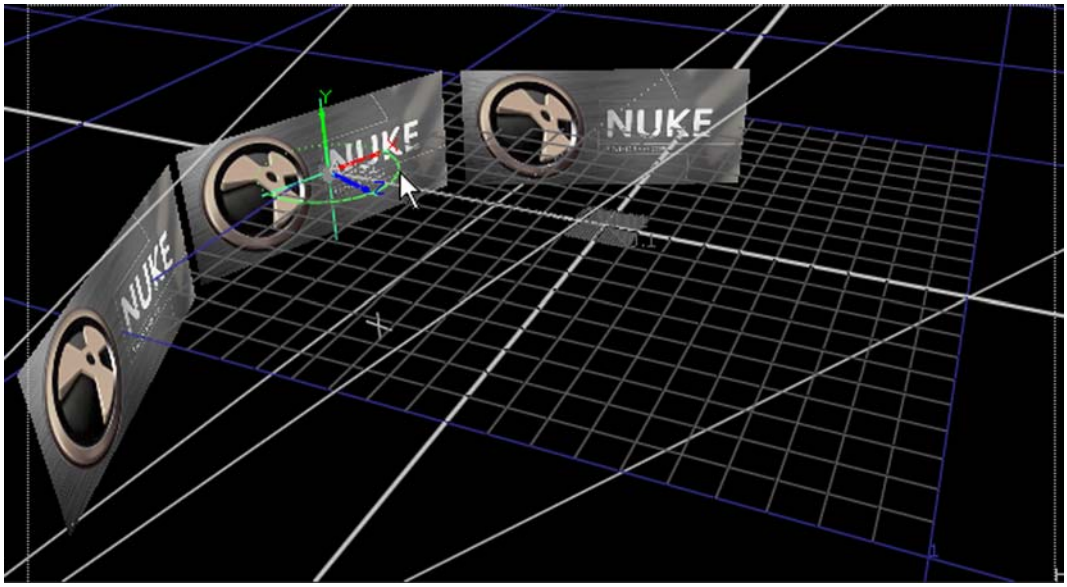
When a camera or object is connected to the optional **look** connector, the TransformGeo node adjusts the rotation so that the object's z-axis always "points" to the camera or object.

The **axis** connector can be used to link the current object to the position, rotation, and scale of a special 3D object called the Axis node. If you've worked with other 3D applications, you know the Axis node as a "null" or "locator" object.

You are still working with the "Setting Up a 3D System" node tree. The following steps show how you can move the merged nodes, and also how to make objects "look" at the camera and other objects.

To move the merged objects together:

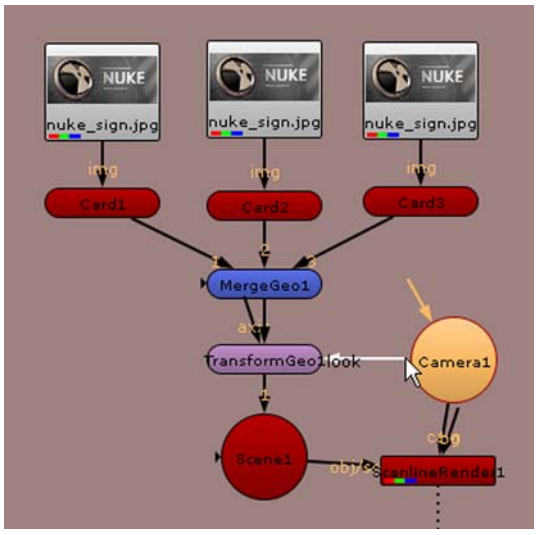
1. Click on the TransformGeo1 node to select it. Its control panel should also be open and you'll see its transform handles in the Viewer.
2. Drag the handles to move all the cards merged with the MergeGeo node.
3. Press the **Ctrl** or **Command** key and drag the rings to rotate the cards as a group.



4. In the **TransformGeo1** control panel, drag the **uniform scale** slider to increase the size of the entire group of cards.

To make objects `look' at the camera:

1. Drag the **look** connector from the **TransformGeo1** node onto the **Camera1** node.



   In the Viewer, you'll now see the TransformGeo1 node is constrained to the location of the camera.

2. Select and move **Camera1** in the Viewer window. As you do so, the three cards controlled by the **TransformGeo1** node rotate to "look at" the camera location.

   Why is this useful? Let's assume you have a 2D matte painting mapped to a card in your scene. The "look" option ensures that the plane of the painting always faces the camera, regardless of the camera position, and maintains the illusion depicted by the painting.

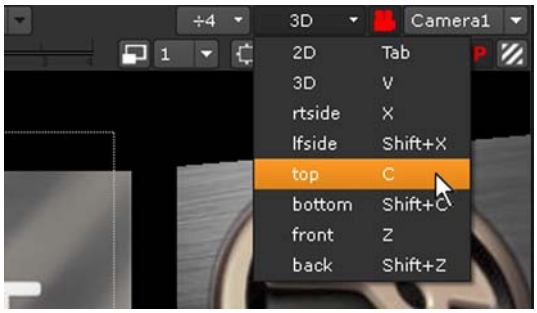
Before you move on, disconnect the TransformGeo node's **look** connector from the camera.

Animating a Scene

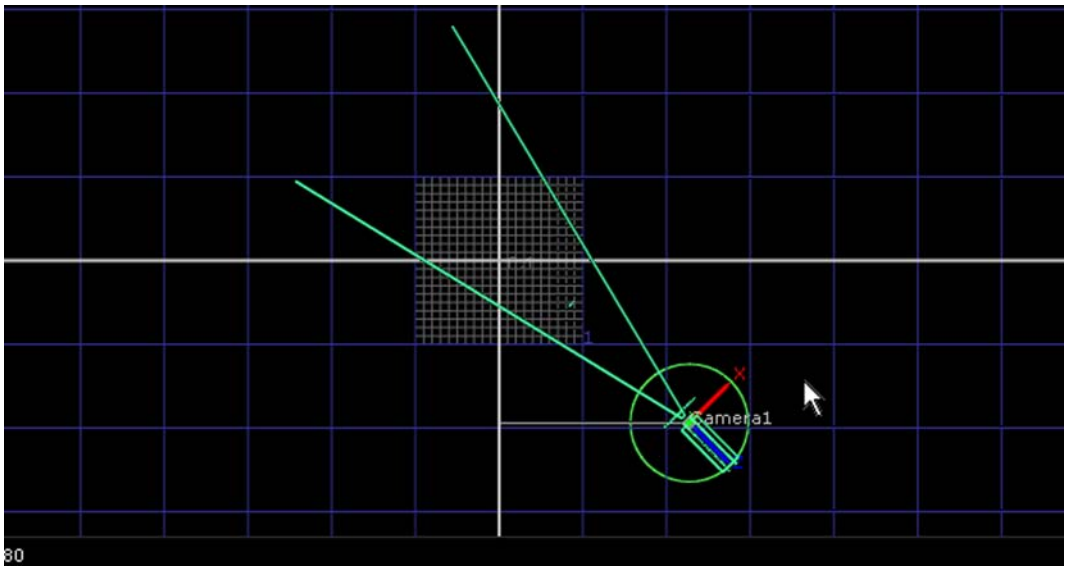
The little scene you've created would be more interesting with a camera move. You can animate both cameras and objects; in each control panel, you'll see an Animation button next to each parameter you can animate over time.

To animate the camera:

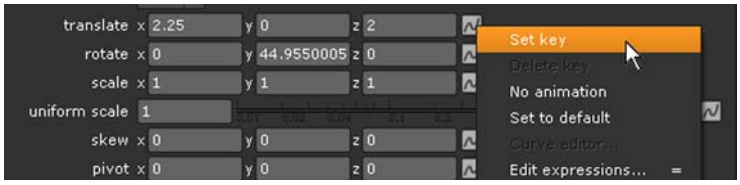
1. In the Viewer, drag the time slider to frame **1** on the timeline.
2. Let's switch to an overhead view to move the camera. Choose **top** from the view list.



3. Double-click on the **Camera1** object (either inside the Viewer or on the node graph) to open its control panel.
4. Move the camera to the right and rotate it to "look" at the centre of the 3D workspace.

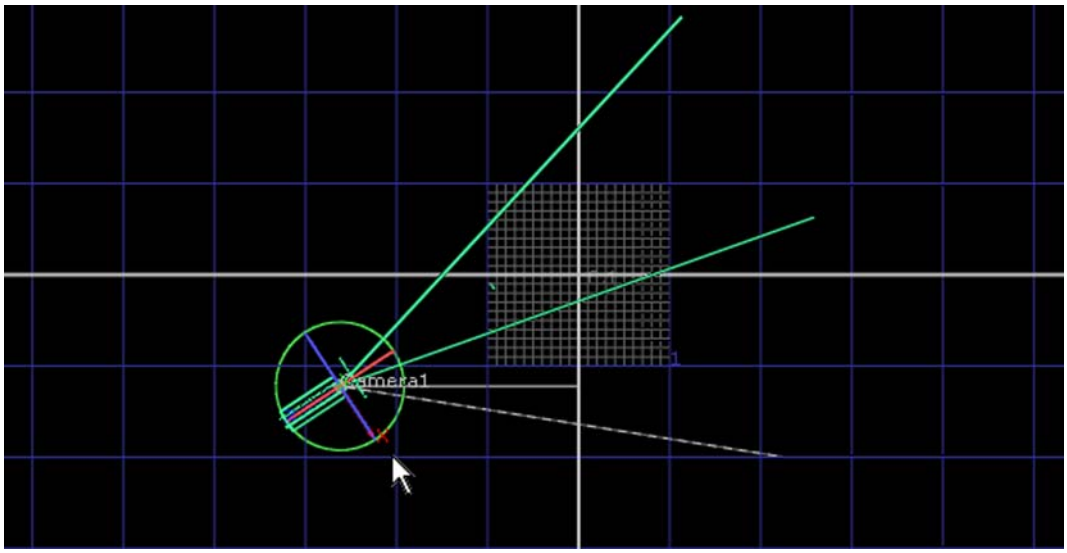


5. Click on the animation button next to the camera's **translate** parameters, and choose **Set key** from the pop-up menu.


translate x 2.25 y 0 z 2
rotate x 0 y 44.9550005 z 0
scale x 1 y 1 z 1
uniform scale 1
skew x 0 y 0 z 0
pivot x 0 y 0 z 0
Set key
Delete key
No animation
Set to default
Curve editor...
Edit expressions...

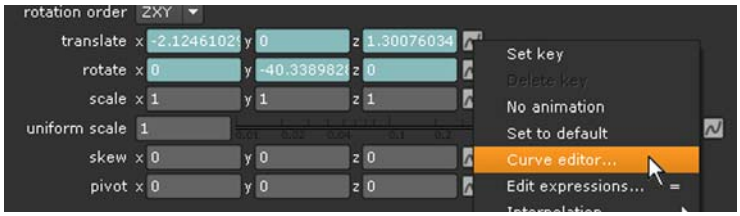
The background of the parameter boxes change colour to show that a key frame is now set for these values at the current frame in the timeline. Now, you need to set a key frame for the rotation values.

6. Next, click the animation button next to the camera's **rotate** parameters, and choose **Set key**.
7. In the Viewer, scrub to the end of the timeline. Then, move the camera to the left side and rotate it to face centre. This automatically sets keys for the **translate** and **rotate** parameters, for the last frame.



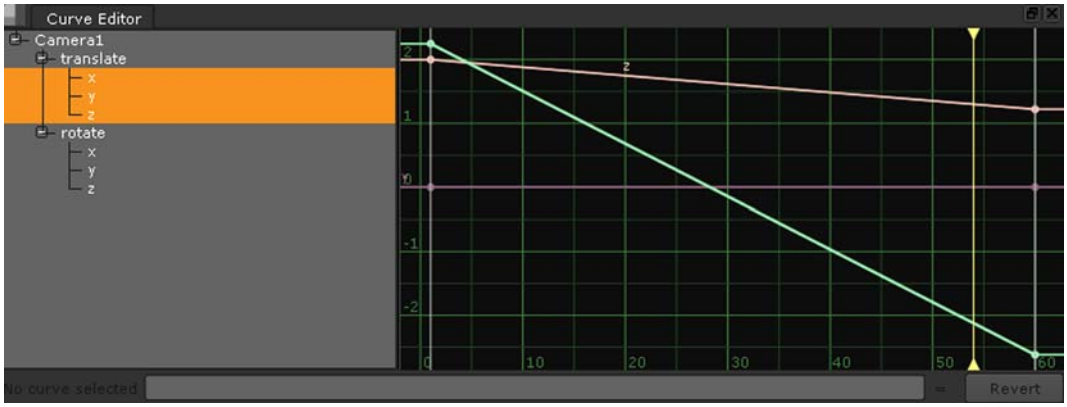
8. Drag through the timeline and the camera moves between the positions recorded by key frames.

   Yawn. With only two key frames, the camera moves in a straight line, from start to finish. Let's edit the animation curves to make this more interesting.
9. Click the animation button next to the camera's translate parameters and choose **Curve editor** from the pop-up menu.


rotation order ZXY
translate x -2.1246102 y 0 z 1.30076034
rotate x 0 y -40.338982 z 0
scale x 1 y 1 z 1
uniform scale 1
skew x 0 y 0 z 0
pivot x 0 y 0 z 0
Set key
Delete key
No animation
Set to default
Curve editor...
Edit expressions...

This opens the Curve Editor window in the same pane as the Node Graph. The outline at the left side of the Curve Editor displays a list of the parameters you've animated, and the

curves show the values plotted over time. Each dot shows the value recorded for a value on a key frame.

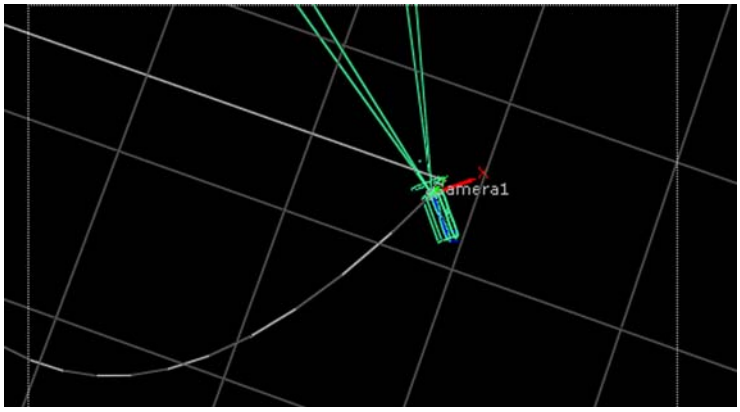


Yes, we know. They don't look like curves - yet. You only set two key frames, remember? You can press **Ctrl+Alt** (Mac users press **Command+Option**) and click on any line to add more key frames. However, let's assume you want to create a smooth arc between the first and last recorded position of the camera. Rather than set more key frames, let's just change the shape of the curve.

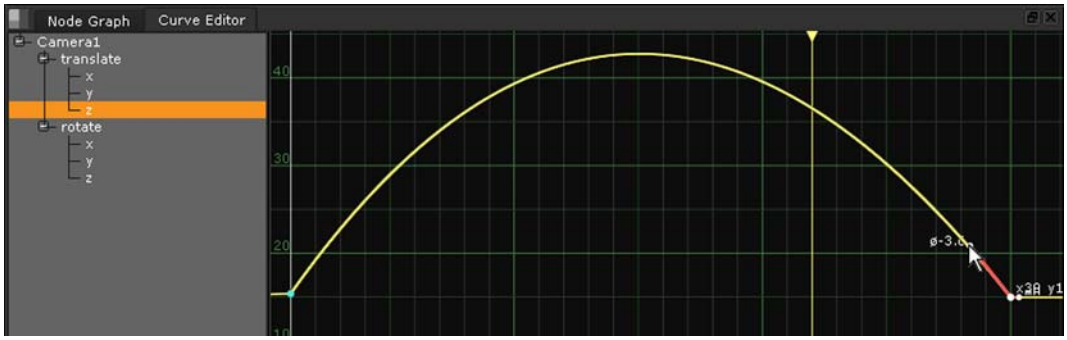
10. You want to control the shape of the camera path along the z—the distance between the origin point and the camera, so click the **translate / z** parameter in the Curve Editor.

    Click on the first point of the translate/z curve to select it, and drag the tangent handle upward, as shown below.

From this point forward, the curve increases the distance of the camera on the z-axis, which you'll now see in the Viewer.

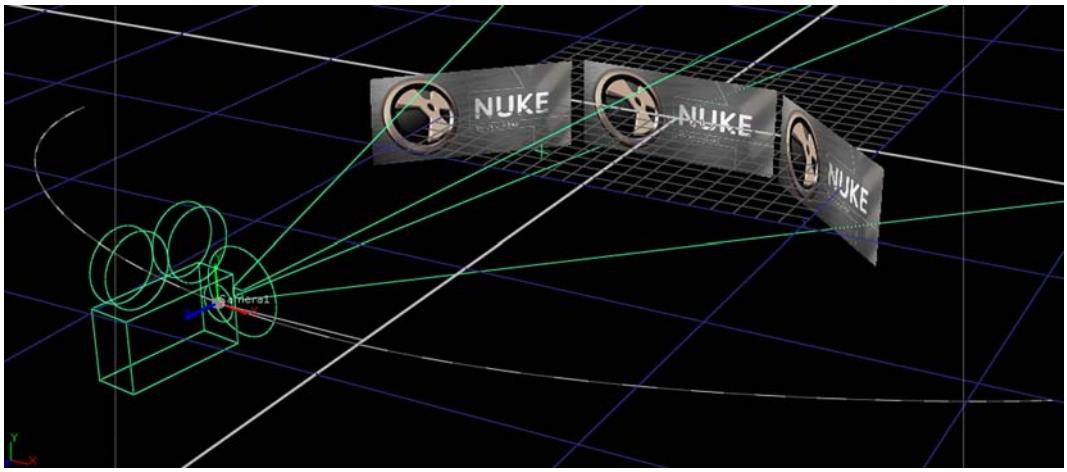


11. Click on the last point of the translate/z curve to select it, and drag it upward also to finish the desired shape. This eases the distance back toward the value of the key frame at the end of the timeline.



Select the camera in the Viewer and you should see the gradual slopes of the curve create a more interesting arc for the camera move.

Switch to the **3D** (**V**) perspective view and scrub through the timeline to see the new camera move.



Your version may look a little different than this, depending on the positions and rotations you defined, but you get the idea.

12. If you wish, you can set additional key frames to refine the camera path. Hold down **Ctrl+Alt** or **Command+Option** and click on the z-axis curve in the Curve Editor, to add new points to the curve, and then adjust their positions.
13. Before you continue, click the **Node Graph** tab to hide the Curve Editor and return to the node trees.
14. Close all the control panels that are currently open.

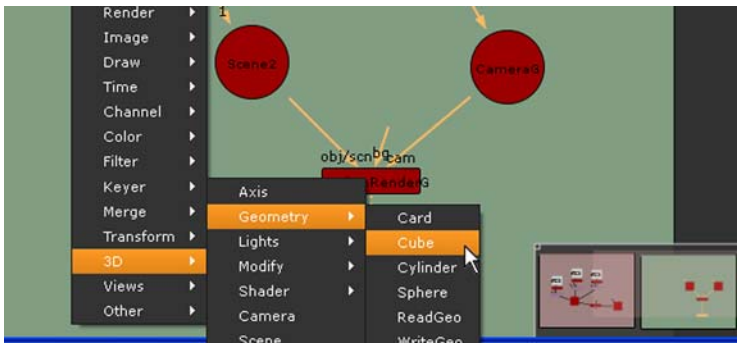
Working with Geometry

In the previous example, you worked with the card object. Nuke also includes primitive geometry, which can be used as set-extension geometry or placeholders for other elements you plan to add to scene.

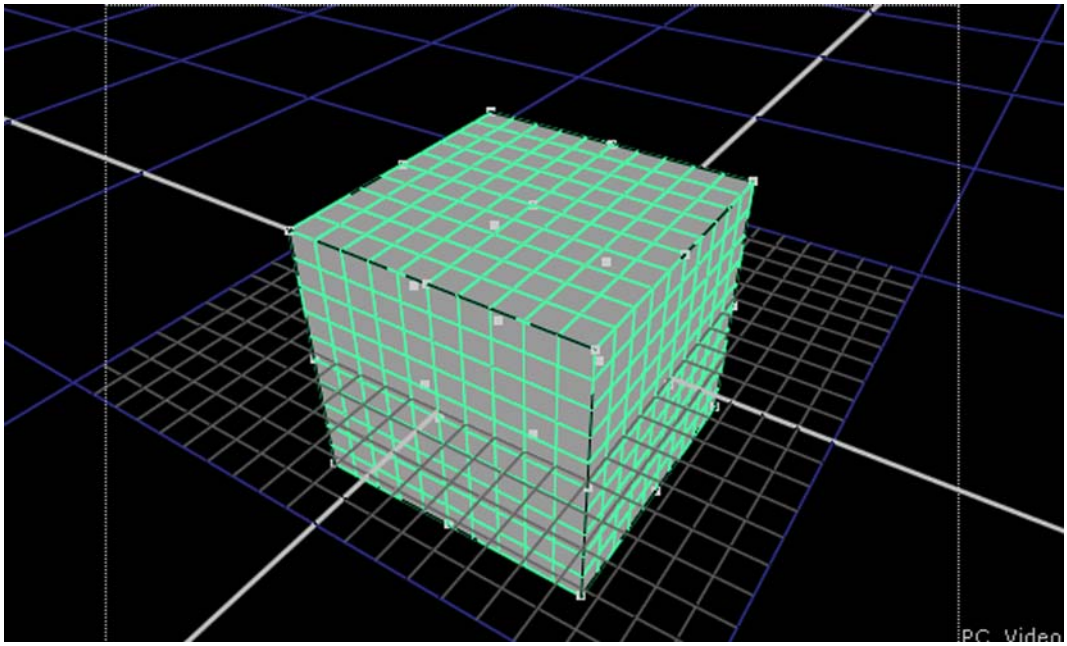
To add primitive objects to the scene:

1. In the "3Dinteg_tutor.nk" project file, locate the node tree labelled "**Working with Geometry**."

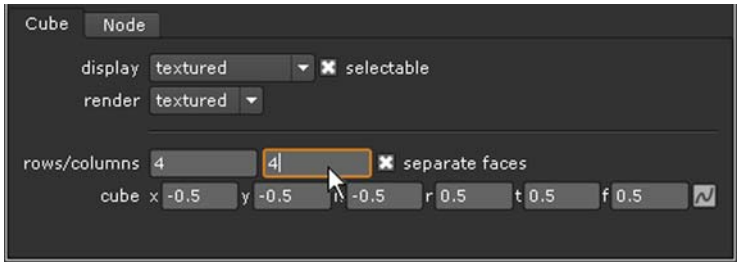
   We've already supplied the 3D node tree with a camera for you, so you need to add the geometry objects, and also create a "scene" where they can coexist.
2. Right-click over the node graph and choose **3D > Scene**. Connect the **Scene2** node to the **ScanlineRenderG** node.
3. Connect a Viewer to the **ScanlineRenderG** node and switch to the **3D** perspective view.
4. Right-click and choose **3D > Geometry > Cube**.



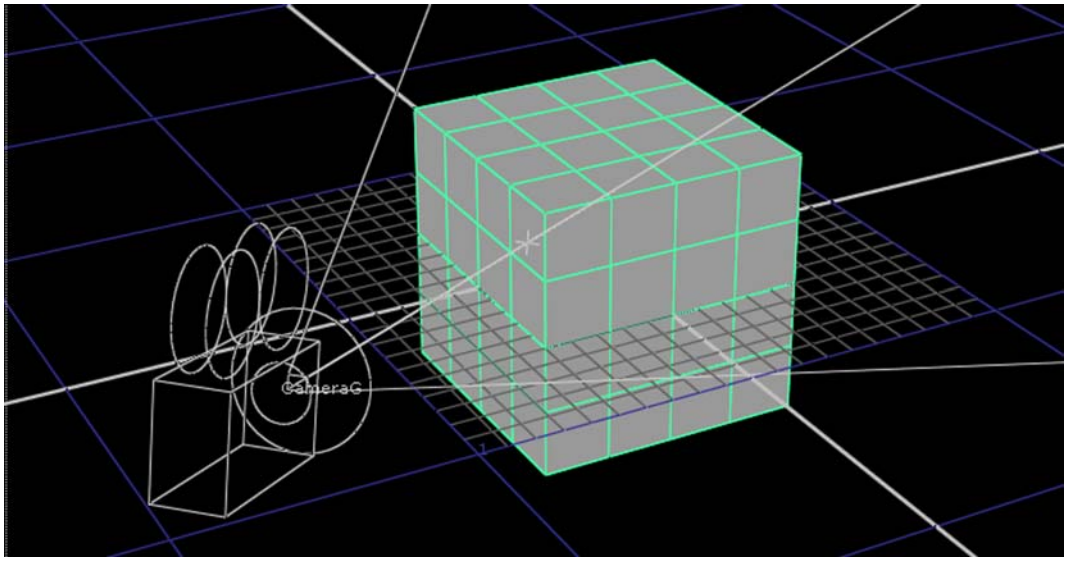
The default cube primitive appears at the centre of the 3D workspace. Let's reduce the number of subdivisions on the cube.



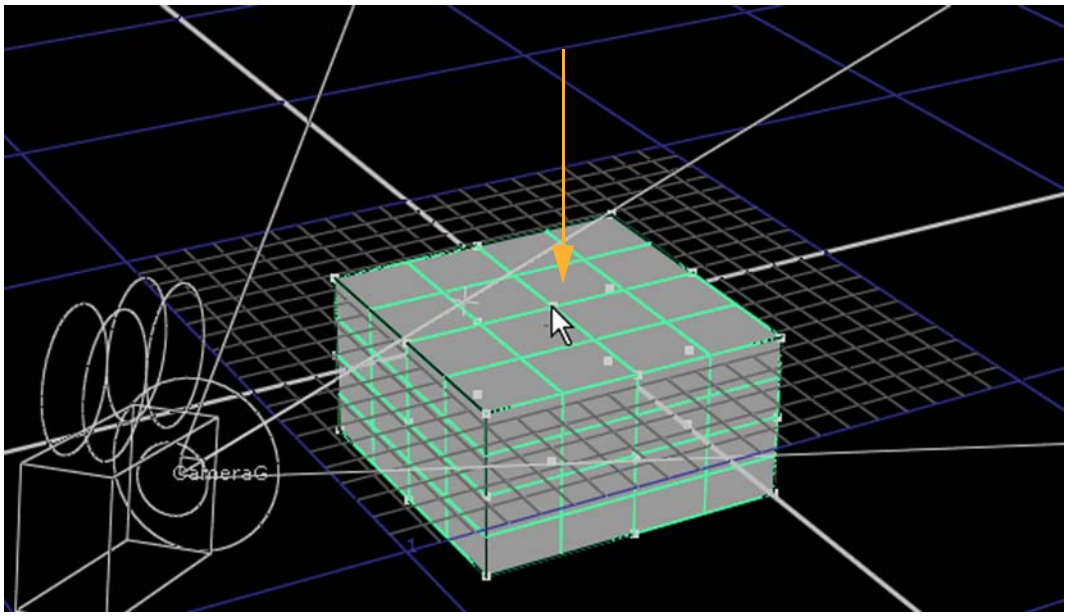
5. In the Cube1 control panel, change the **rows** parameter to **4**. Change the **columns** parameter to **4**, also.



6. Connect the **Cube1** node to the **Scene2** node. Now let's adjust the shape of the cube.



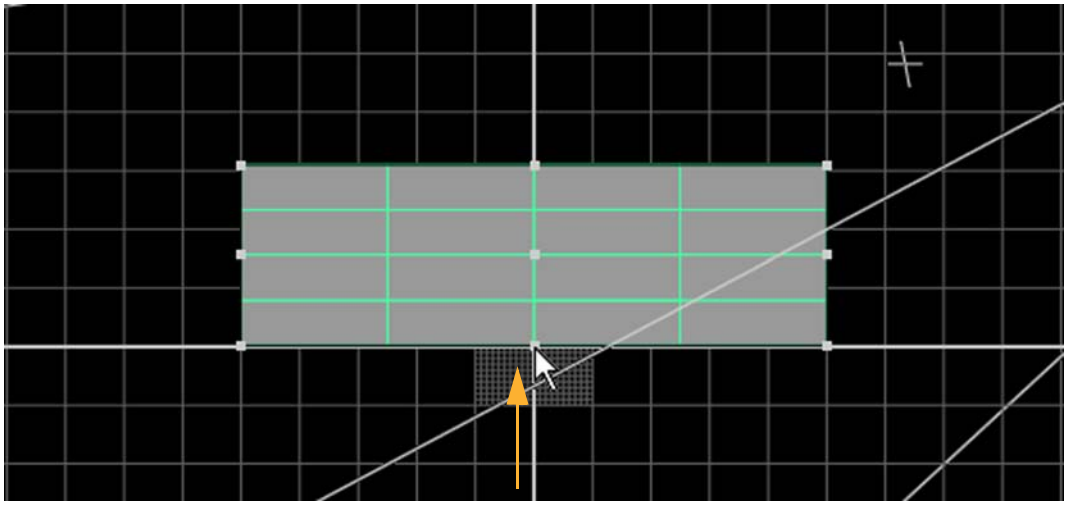
7. Reduce the height of the cube by dragging the top-centre point down.



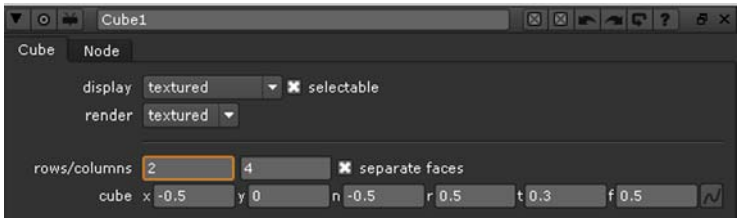
8. From the view list, choose the **front** view to see a non-perspective view of the cube. These non-perspective views can help you size and position objects with more accuracy than you might get in a perspective view.

   Mm... the cube is actually below the x-axis. Let's move it up, but check the values in the **Cube1** control panel.

9. Drag the top of the cube until the **t** (top) value in the **Cube1** control panel is about **0.3**. Drag the bottom of the cube to align it with the x-axis.

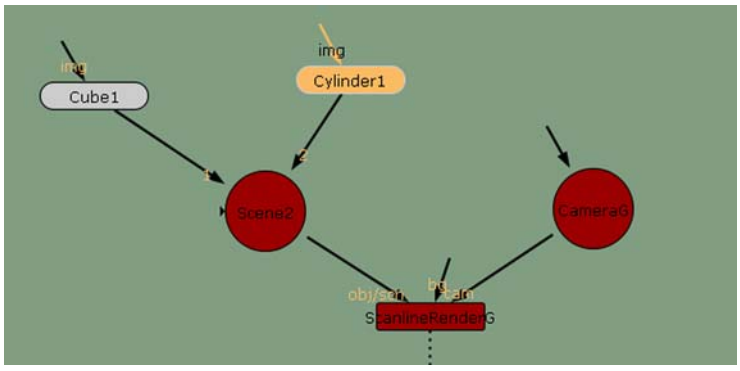


10. It looks like you don't need 4 divisions on the sides of the cube, so change the number of rows to **2** in the **Cube1** control panel.

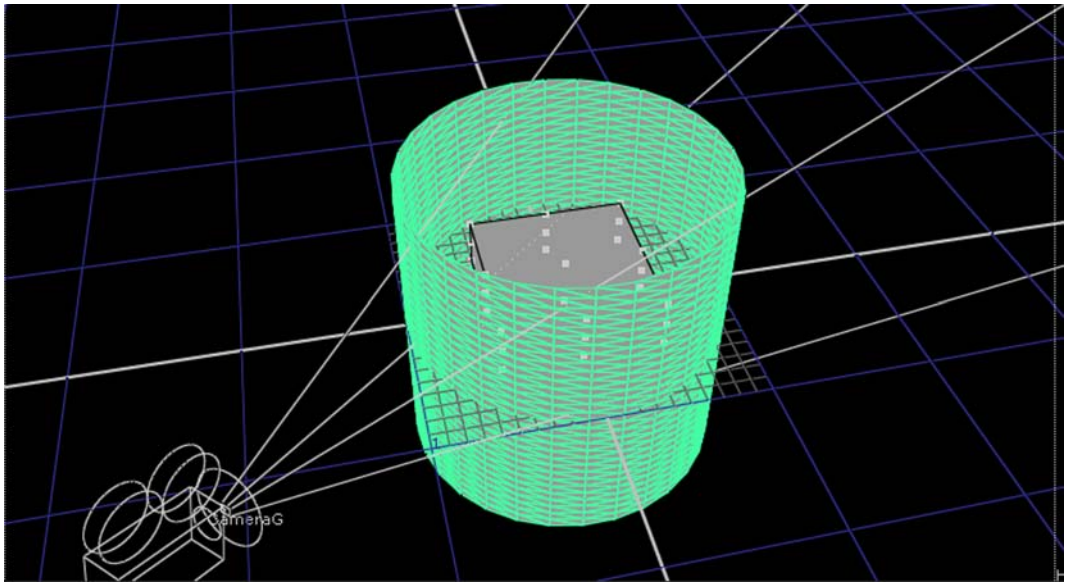


Now let's add a few more primitives—a cylinder and a sphere.

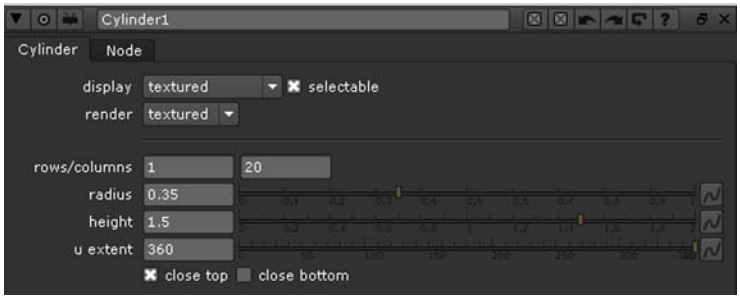
11. Right-click on the node graph and choose **3D > Geometry > Cylinder**. Connect the **Cylinder1** node to the **Scene2** node.



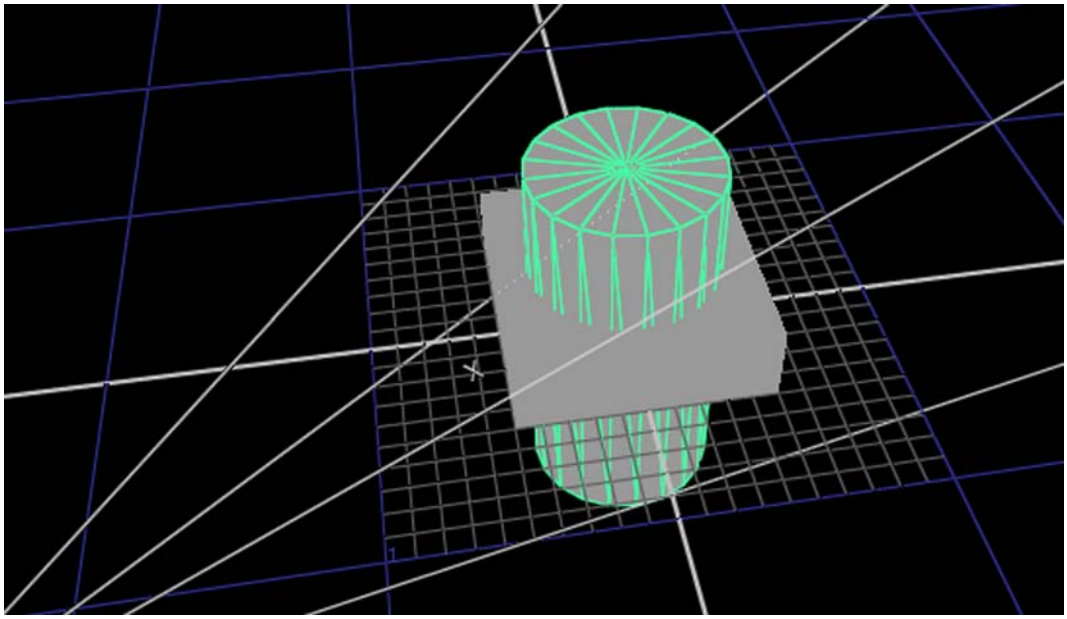
12. Change the view to **3D** (**V**) and zoom out a little to see the whole cylinder.



13. In the Cylinder1 control panel, set the **rows** to **1**, the **columns** to **20**.

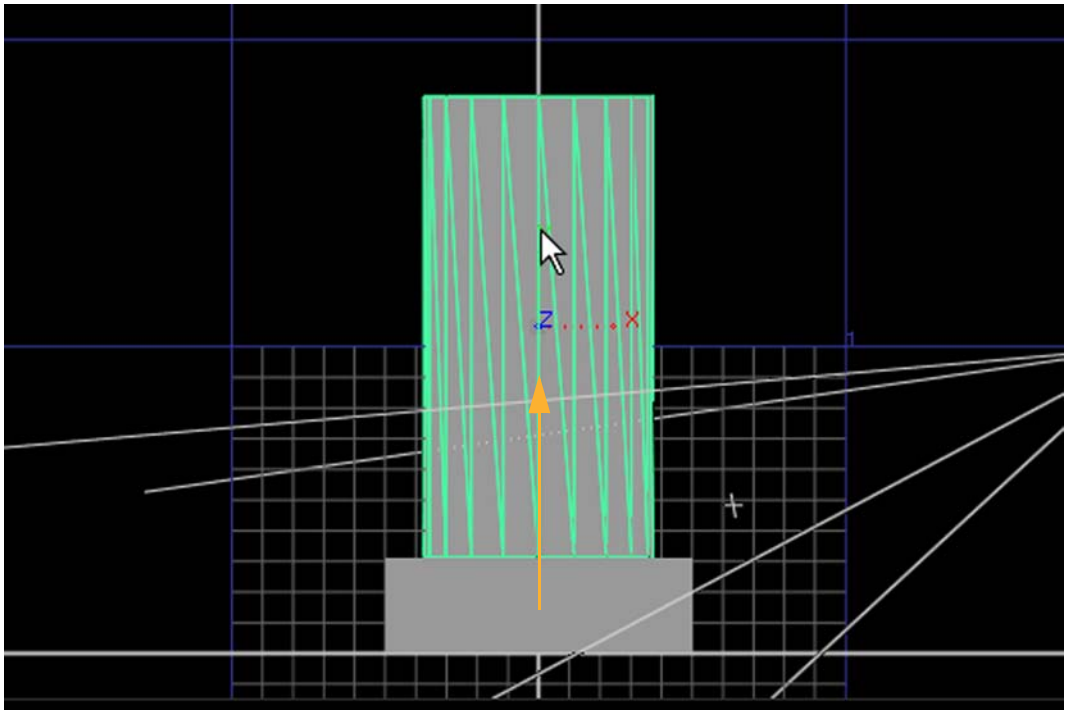


14. Set the **radius** to **0.35** and the height to **1.5**. Also check the box for **close top**.

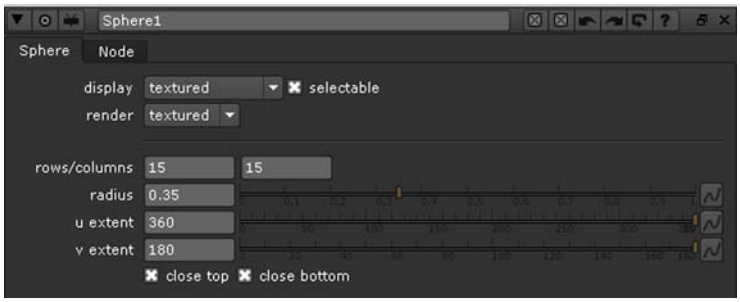


So now you have a cylinder in the scene.

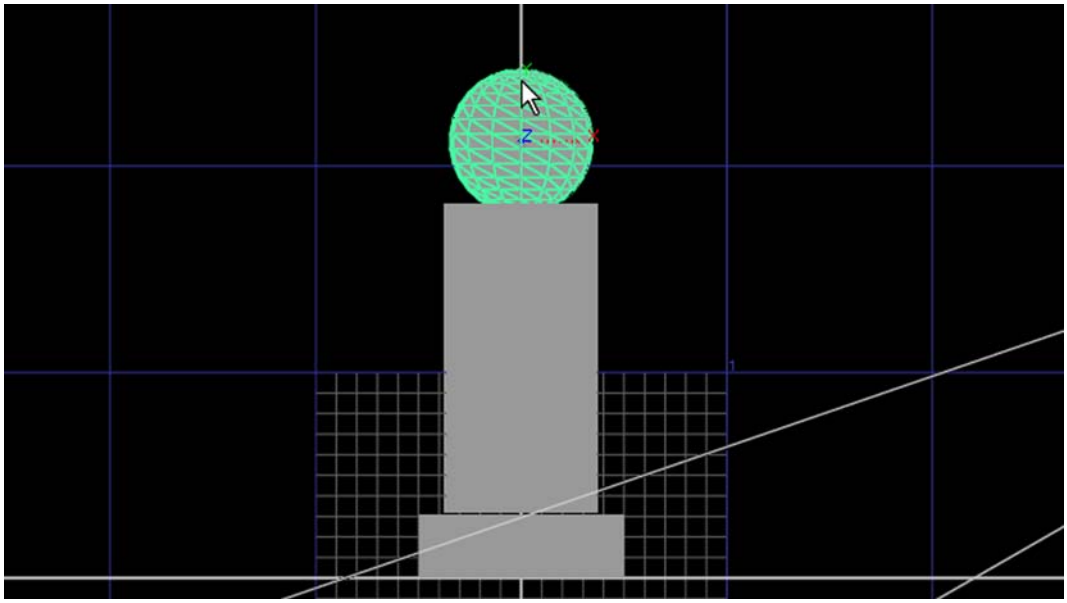
15. Choose **front** from the view list and move the cylinder up to rest on top of the cube.



16. Now add a sphere. Choose **3D > Geometry > Sphere**. In the Sphere1 control panel, set both the **rows** and **columns** to **15**, and change the **radius** to **0.35**.



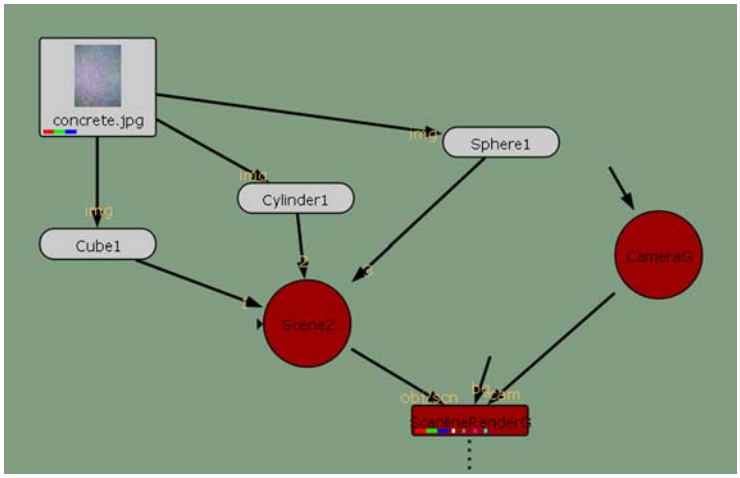
17. Make sure the Sphere control panel is open and move the sphere object to cap the top of the cylinder.

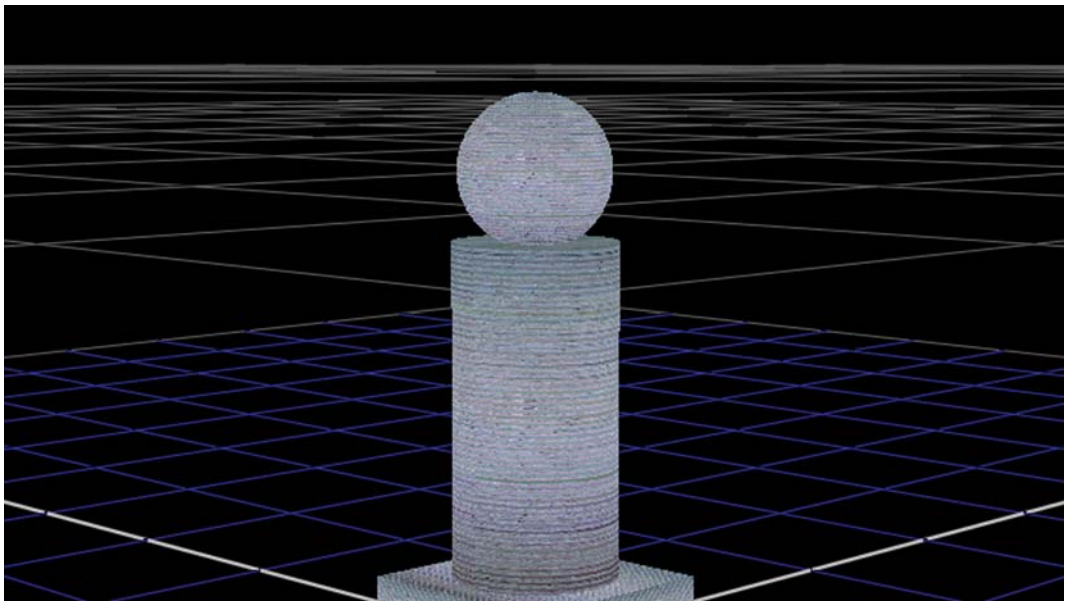


18. Select the **3D** from the view list and rotate the view around the objects in your scene.

    At this point, they have no surface properties, so you'll need to connect a 2D image from the node graph to each object.

19. In the node graph, connect the **concrete.jpg** to each of the objects.

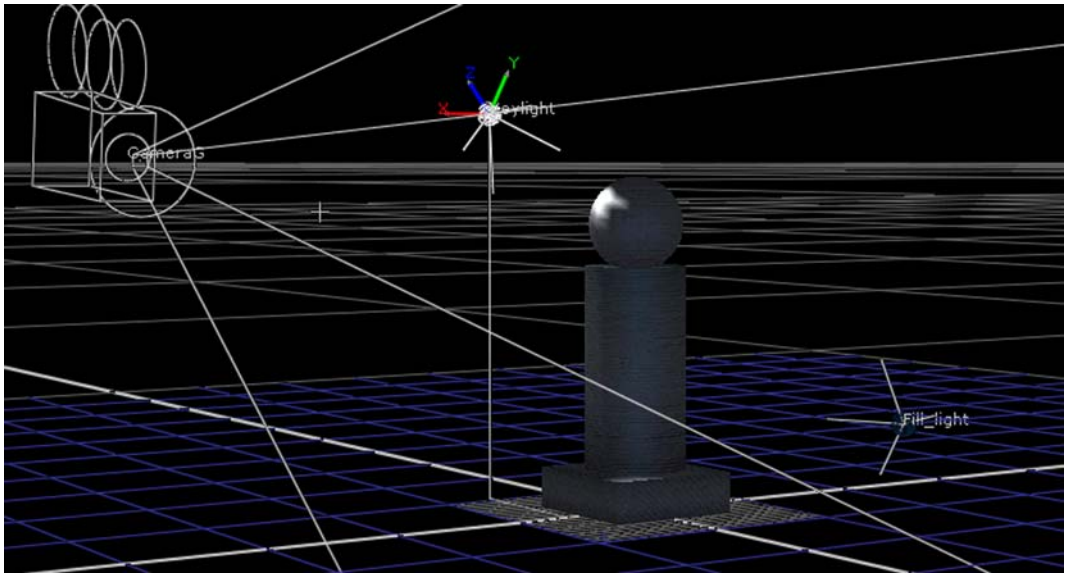




Lighting & Surface Properties

Nuke includes lighting tools to enhance the existing lighting in the plates and images you include in a 3D scene. Also included are fundamental surfacing tools to control the attributes of the objects in the 3D workspace.

These tools are not designed to replace the use of true 3D lighting and surfacing, but they can definitely help you punch up the scene and quickly tweak the settings without sending elements back to the 3D application.

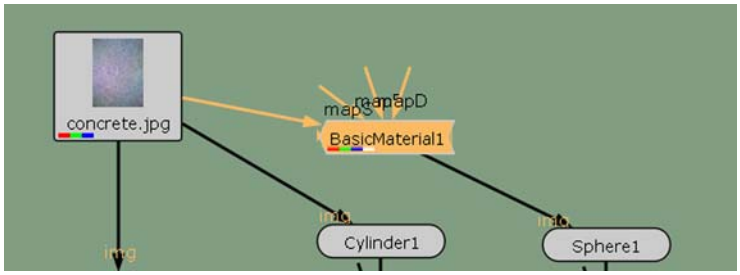


Nuke's lighting objects introduce lighting and surface attributes into your scene. When the scene has no lighting objects, then all surfaces are illuminated with the same properties and level of "brightness."

In the following steps, you'll first add nodes that define surface properties for the objects, and then you'll add the light objects to illuminate them.

To define surface attributes to objects:

1. Click on an empty place in the node graph to deselect all nodes. Then, choose **3D > Shader > BasicMaterial**.
2. Drag the **BasicMaterial1** node onto the connector between concrete.jpg and **Sphere1**.

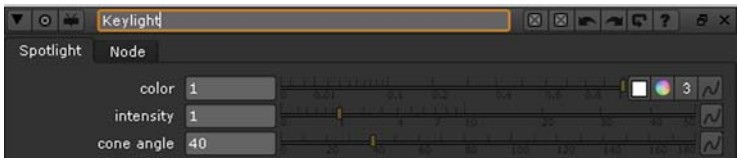


In the Basic Material control panel, you will see parameters to define the amount of light emission, diffusion, and specular properties of the surface. You can mask these properties by connecting images to the **mapS** (specular), **mapE** (emission), and **mapD** (diffuse) connectors, but this is not required for this example.

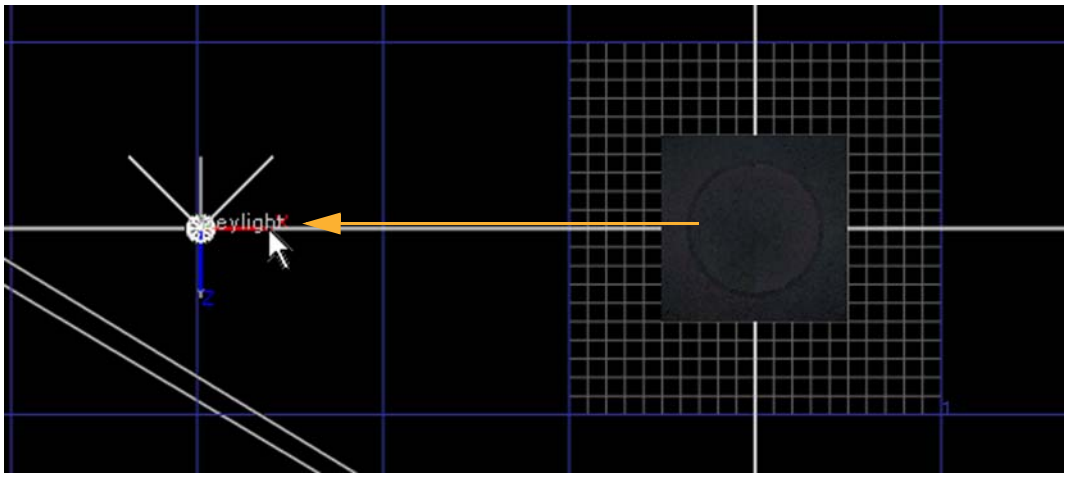
3. Set the light **emission** control to **0.25**. Set **diffuse** to **0.18**, and **specular** to **0.75**.
4. Adjust the **min shininess** and **max shininess** values to adjust the quality of the specular highlights. Once again, nothing seems to happen! That's because you haven't yet added a light into the scene. All surfaces are illuminated with the same level of brightness, so there are no specular highlights or other controllable light properties visible. It's not very exciting, but don't worry - you get to add a light into the scene soon.
5. Make two copies of the **BasicMaterial1** node and attach a copy before the **Cylinder1** node and before the **Cube1** node.

To add light objects to a scene:

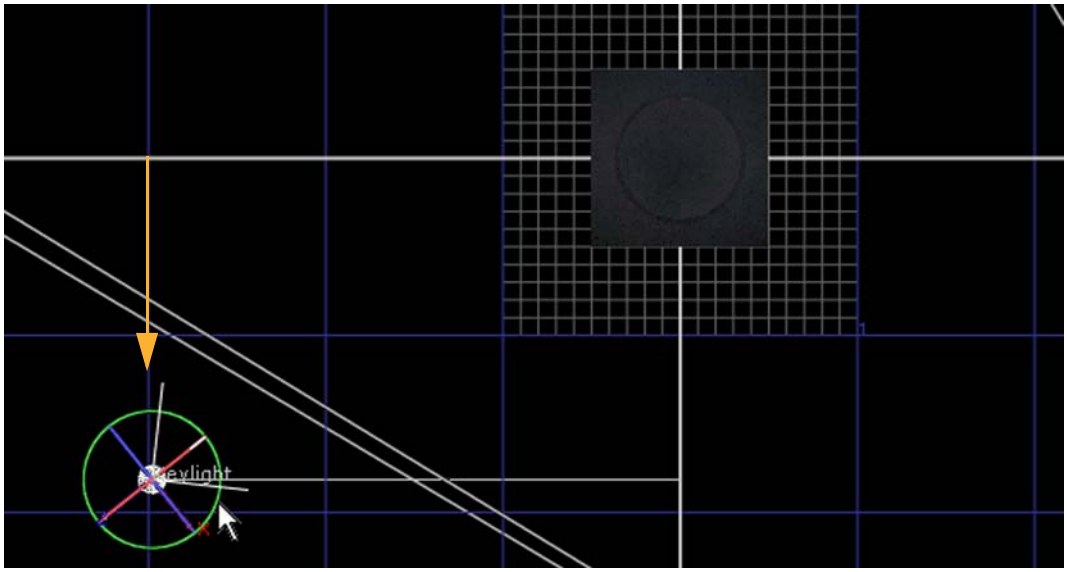
1. Choose **3D > Lights > Spot** and connect the light node to the **Scene2** node.
2. In the **Spotlight1** control panel, rename the light to **Keylight**.



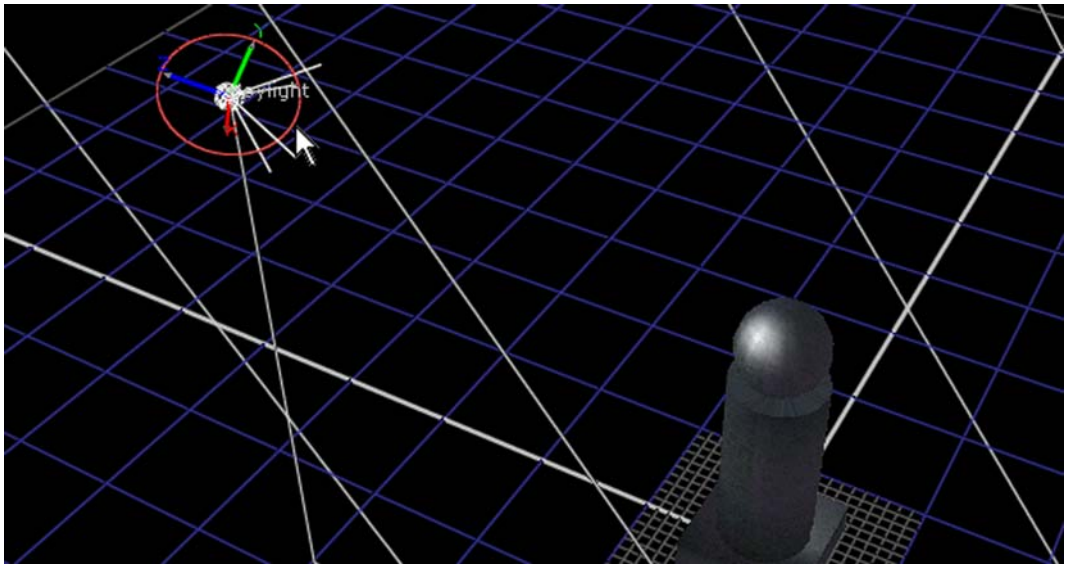
3. Switch to the **top** view and drag x-axis handle (red) to move the light to the left.



4. Drag the z-axis handle (blue) to move the light closer to the bottom edge of the screen. Then, press the **Ctrl** or **Command** key and rotate the light to face the pillar object.



5. Switch to the **3D** (V) view and rotate the view so you can see both the Keylight and the pillar geometry.
6. Drag the y-axis handle (green) to move the light up above the pillar.
7. Press the **Ctrl** or **Command** key and rotate the light down to shine on the pillar.



That's the basic setup for lighting and surfaces, but there are other tools for more complex setups. Refer to the Nuke user guide for more information on the 3D lighting and surfacing tools.

Epilogue

In this chapter, you learned how to setup a 3D workspace for your composite, and how to work with 3D geometry and lighting. These are all prerequisite skills for more advanced topics, such as camera projections, matchmoving, and set replacement.

- End of Tutorial -

APPENDICES

This section contains supplemental reference information that you may need when using Nuke.

Organisation of the Section

The section consists of four appendices:

- *Appendix A: Hot Keys* lists the keyboard shortcuts you can use for quicker and easier access to Nuke's features. You can also open a list of keyboard shortcuts from the application by selecting **Help** > **Key Assignments**.
- *Appendix B: Supported File Formats* lists the image and video file formats Nuke supports.
- *Appendix C: Converting from Shake to Nuke* shows you the main differences between Apple's Shake and Nuke. If you are familiar with Shake but not yet Nuke, we recommend you read this appendix.
- *Appendix D: Third Party Licenses* lists third party libraries used in Nuke, along with their licenses.
- *Appendix E: End User Licensing Agreement* shows you the End User License Agreement that governs the use of Nuke software and this user guide.

Appendix A: Hot Keys

Keystroke shortcuts, or hot keys, provide quick access to the features of Nuke. The following tables show these keystrokes.

This appendix assumes you are using the default keyboard and mouse-button assignments. If the mouse buttons do not work for you as described here, try resetting the mouse control type back to the standard Nuke setting (**Edit > Preferences > Viewers > 3D Control Type = Nuke**).

Conventions

The following conventions apply to instructions for mouse-clicks and key presses.

- LMB means click or press the left mouse button.
- MMB means click or press the middle mouse button
- RMB means click or press the right mouse button.
- When you see the word "drag" after a mouse button abbreviation (i.e., "**MMB drag**"), this tells you to press and hold the mouse button while dragging the mouse pointer.
- Keystroke combinations with the **Ctrl**, **Alt**, and **Shift** keys tell you to press and hold the key and then type the specified letter.
  For example, "Press **Ctrl+S**" means hold down the **Ctrl** key, press **S**, and then release both keys.

Important

On Mac OS X, replace the **Ctrl** key with the **Cmd** key.

Important

Keystrokes in the tables appear in upper case, but you do not type them as upper case. If the **Shift+** modifier does not appear before the letter, just press the letter key alone.

Node Graphs, Viewers, Curve Editors, Script Editors, and Properties Bins

Keystroke(s)	Action
/	Search by node name or class.
Alt+G	Go to a specific frame.
Alt+I	Display script information, such as the node count, channel count, cache usage, and whether the script is in full-res or proxy mode.
Alt+S	Make the active (floating) window fullscreen.
Alt+Shift+S	Save script and increment version number. (See also **Shift**+**Ctrl**+**S** under *Viewers*.)
Ctrl+F#	Save current window layout. The # represents a function key number, **F1** through **F6**.
Ctrl+I	Open new Viewer window.

Keystroke(s)	Action
Ctrl+LMB on panel name	Float panel.
Ctrl+N	Launch a new project window in a new instance of Nuke.
Ctrl+O	Open a script file.
Ctrl+Q	Exit Nuke.
Ctrl+Y	Redo last action.
Ctrl+Z	Undo last action.
Shift+Ctrl+S	Save script and specify name (Save As). (See also **Alt**+**Shift**+**S** under *Viewers*.)
Shift+S	Open Nuke Preferences dialog.
Space bar (short press)	Expand the focused panel to the full window.
Space bar (long press)	Raise the right-click menu.

Properties panels

Keystroke(s)	Action
up or down arrow	Increment (up) or decrement (down) the value in a parameter field. Click first on the field or press Tab to move focus to the parameter.
Alt+LMB drag	Increment (drag left) or decrement (drag right) while dragging over the value in a parameter field.
Alt+R *or* Ctrl+R	Fit Properties Bin to open panels.
Ctrl+A	Select all nodes in the Properties Bin.
Ctrl+Enter (NUM)	Close current panel.
Ctrl+LMB	Reset slider value to default.
Ctrl+LMB drag	Link values between parameter fields.
Ctrl+Return	Close panel (no parameters selected).
Ctrl+Tab *or* Shift+Ctrl+Tab	Move to next tabbed page (**Ctrl+Tab**) or the previous tabbed page (**Shift+Ctrl+Tab**) in the properties panel.
LMB drag	Copy the current value from one parameter field to another.
Return	Chooses selected UI control (default = **OK**).
Shift+Ctrl+A	Close all open properties panels.
Shift+LMB drag	Copy animation (curve or expression) from one parameter field to another.
Tab *or* Shift+Tab	Move focus to next (**Tab**) or previous (**Shift+Tab**) parameter. May need to click on a parameter first, to establish the focus inside the properties panel.

Node Graph

Keystroke(s)	Action
+	Zoom-in (**=** also zooms-in).
-	Zoom-out.
\	Snaps all nodes to the grid. (See also **Shift**+\ under *Node Graph*.)
#	Opens a new Viewer window with **#** representing the number of the connection (**0** to **9**) you want to establish between the new Viewer and the selected node.
.	Inserts Dot node.
up or down arrow	Selects the previous or next node in the tree.
Alt+up arrow	Increment the version number in the selected node's filename.
Alt+down arrow	Decrement the version number in the selected node's filename.
Alt+#	Zoom-out by a specific percentage. The # represents a number between **0** and **9**, with **0**=10%, **1**=100%, **2**=50%, **3**=30%, **4**=25%, **5**=20%, **6**=16%, **7**=14%, **8**=12%, and **9**=11%.
Alt+B	Duplicate and branch selected nodes.
Alt+C	Duplicate selected nodes.
Alt+F	Generate flipbook for selected node using FrameCycler.
Alt+K	Clone selected nodes.
Alt+LMB drag	Pan workspace.
Alt+MMB drag	Zoom-in / zoom-out workspace.
Alt+Shift+K	Remove selected nodes from clone group (declone).
Alt+Shift+U	Splay last selected node to input A.
Alt+U	Splay first selected node to input A.
B	Insert Filter Blur node.
Ctrl	Display connector dots. Drag one to set a dot and create an "elbow."
Ctrl+down arrow	Move selected node downstream.
Ctrl+up arrow	Move selected node upstream.
Ctrl+#	Zoom-in by a specific percentage. The # represents a number between **0** and **9**, with **0**=1000%, **1**=100%, **2**=200%, **3**=300%, **4**=400%, **5**=500%, **6**=600%, **7**=700%, **8**=800%, and **9**=900%.
Ctrl+Alt+LMB on a node	Open the node's properties panel in a floating window.

Keystroke(s)	Action
Ctrl+A	Select all nodes in the Node Graph or group window.
Ctrl+B	Node buffer toggle.
Ctrl+C	Copy selected nodes.
Ctrl+D	Disconnect upstream node from selected node.
Ctrl+G	Nest selected nodes inside a Group node, keeping the original nodes in the layout. **Ctrl+Enter** opens the new Group node.
Ctrl+P	Toggle proxy resolution, as defined on the Settings properties panel. (See also **Ctrl+P** under *Viewers*.)
Ctrl+Return	Open window for selected group node.
Ctrl+Shift+/	Opens the Search and Replace dialog for the selected Read or Write nodes.
Ctrl+W	Close current script file.
D	Disable / enable selected node.
Delete	Remove selected nodes.
F	Fit the selected nodes (or if no nodes are selected, the entire node tree) to the Node Graph panel or group window.
F12	Clear buffers.
I	Display information for selected node.
K	Insert Copy node. Note that for this to work, you first need to click on the Node Graph to select it. If you have selected the Viewer, pressing K stops playback. (See **K** under *Viewers*.)
M	Insert Merge node.
N	Rename the selected node.
P	Insert Bezier node.
Return	Open panel for selected node(s).
Shift+\	Snaps selected node to the grid. (See also **** under *Node Graph*.)
Shift+0, 1, 2, 3, ...	Connect the selected node to Viewer as reference input.
Shift+drag	Duplicate selected arrow.
Shift+A	Insert AddMix node.
Shift+Ctrl+C	Set colour for selected nodes.
Shift+Ctrl+X	Extract selected nodes.
Shift+U	Splay selected nodes to last selected node.

Keystroke(s)	Action
Shift+X	Swap A/B inputs on selected node.
Tab	Open a text field where you can enter the first letters of a node name to bring up a list of matches. Press Return to insert a node from the list.
U	Splay selected nodes to first selected node.
Y	Connect selected node to second selected node.

Editing

Keystroke(s)	Action
Alt+Ctrl+V	Paste knob values.
Backspace	Erase or Delete Left.
Ctrl+B	Left justify selected text.
Ctrl+C	Copy.
Ctrl+E	Move cursor to end of selected text.
Ctrl+F	Right justify selected text.
Ctrl+K	Delete text from the cursor to the next space.
Ctrl+N	Bottom justify selected text.
Ctrl+P	Top justify selected text.
Ctrl+V	Paste.
Ctrl+X	Cut.
Shift+Ctrl+V	Paste (see **Ctrl+V**).

Viewers

Keystroke(s)	Action
-	Zoom-out.
+	Zoom-in (= also zooms-in).
.	Gain display, increase.
; (semicolon)	Switch to the previous view in a multiview project.
` (forward single quote)	Switch to the next view in a multiview project.
!	Turn on Viewer "blend," split-screen display. (See also **W** under *Viewers* for toggle on/off).

Keystroke(s)	Action
[	Hide all Viewer controls.
0, 1, 2, 3, ...	Establish a numbered connection (**1** - **9**, **0**) between the selected node and the active Viewer. Displays the node's output in that Viewer.
Shift+0, 1, 2, 3, ...	Connect reference inputs to Viewer.
Numeric keypad	Nudge on-screen controls left, right, up, or down. (See also **Shift+numeric keypad** under *Viewers*.)
Right arrow	Step forward one frame.
Left arrow	Step back one frame.
, (comma)	Gain display, decrease.
` (accent key)	Show / hide all Viewers.
A	Display the alpha channel or the channel displayed in the list at the top of the Viewer.
Alt+left arrow	Previous key frame.
Alt+right arrow	Next key frame.
Alt+*#*	Zoom-out by a specific percentage. The # represents a number between **0** and **9**, with **0**=10%, **1**=100%, **2**=50%, **3**=30%, **4**=25%, **5**=20%, **6**=16%, **7**=14%, **8**=12%, and **9**=11%.
Alt+G	Go to specific frame.
Alt+LMB drag	Pan inside the Viewer window.
Alt+MMB drag	Zoom in (drag right) or out (drag left) in the Viewer window.
Alt+P	Activate the VIEWER_INPUT node.
Alt+R	Resize Viewer to image (see also **Ctrl+R** under *Viewers*).
Alt+Shift+R	Resize Viewer and image to fit frame.
Alt+W	Activate the ROI feature, or if a ROI exists, clear the current ROI. Drag to define a new ROI. (See also **Shift+W** under *Viewers*.)
B	Display blue channel / RGB toggle.
Backspace	Cycle through Viewer inputs in reverse order. If wipe is active, cycles through inputs on the left-hand side.
Ctrl+right arrow	Move to midpoint between current frame and next key frame/last frame.
Ctrl+left arrow	Move to midpoint between current frame and previous key frame/first frame.
Ctrl+#	Zoom-in by a specific percentage. The # represents a number between **0** and **9**, with **0**=1000%, **1**=100%, **2**=200%, **3**=300%, **4**=400%, **5**=500%, **6**=600%, **7**=700%, **8**=800%, and **9**=900%.

Keystroke(s)	Action
Ctrl+Alt+LMB	Sample a single pixel's colour value from the node's input while viewing its output. (See also **Ctrl**+**LMB** under *Viewers*.)
Ctrl+Alt+Shift+LMB	Sample range of pixels from the node's input while viewing its output. (See also **Ctrl**+**Shift**+**LMB** under *Viewers*.)
Ctrl+LMB	Sample a single pixel's colour value from the Viewer. (See also **Ctrl**+**Alt**+**LMB** under *Viewers*.)
Ctrl+P	With the mouse pointer over the Viewer, this keystroke toggles pixel aspect ratio between square and non-square, according to the setting of the default format under the Settings properties panel. This is not the same as toggling proxy resolution (see also **Ctrl+P** under *Node Graph*).
Ctrl+R	Resize Viewer window to image (see **Alt+R** under *Viewers*).
Ctrl+Shift+LMB	Sample range of pixels from the Viewer. (See also **Ctrl**+**Alt**+**Shift**+**LMB** under *Viewers*.)
End	Go to last frame.
Esc	Close Viewer.
F	Fit image to Viewer with bounding box border and output labels.
G	Display green channel / RGB toggle.
H	Fill image in Viewer.
J	Play backward.
K	Stop playback. Note that for this to work, you first need to click on the Viewer to select it. If you have selected the Node Graph, pressing K inserts a Copy node. (See **K** under *Node Graph*.)
L	Play forward.
M	Display Matte, or alpha channel as transparent overlay.
MMB	Fit Viewer to frame.
O	Show / hide overlays.
P	Disable (pause) the display refresh of the Viewer.
R	Display red channel / RGB toggle.
RMB (or press and hold the spacebar)	Display Viewer menu.
S	Display Viewer Settings dialog.
Shift+left arrow	Move left on the timeline by the specified increment amount.
Shift+right arrow	Move right on the timeline by the specified increment amount.

Keystroke(s)	Action
Shift+A	Display "other" channel / RGB toggle. Current input only. (Default = Alpha).
Shift+B	Display Blue channel / RGB toggle. Current input only.
Shift+Backspace	Activate Wipe. Cycle images on right side.
Shift+Ctrl+R	Resize Viewer to maximum and fit image.
Shift+F	Maximum Viewer window toggle.
Shift+G	Display Green channel / RGB toggle. Current input only.
Shift+L	Display Luminance / RGB toggle. Current input only.
Shift+M	Display Matte / RGB toggle. Current input only.
Shift+numeric keypad	Nudge on-screen controls left, right, up, or down by increment. (See also **Numeric keypad** under *Viewers*.)
Shift+R	Display Red channel / RGB toggle. Current input only.
Shift+W	Region of interest (ROI) toggle. (See also **Alt+W** under Viewers.)
Tab	2D / 3D view toggle.
U	Update Viewer display. Used when Pause is active (press **P**).
W	Toggles Viewer "blend," split-screen display, on/off. This was previously called the "wipe" mode. (See also **!** under *Viewers*).

3D Viewer

Keystroke(s)	Action
Alt+LMB	Translate Viewer perspective on y (drag up/down) or z (drag left/right).
Alt+MMB	Zoom Viewer perspective in (drag right) or out (drag left).
Alt+RMB	Rotate Viewer perspective on x (drag up/down) or y (drag left/right).
Ctrl+LMB	Rotate Viewer perspective on x (drag up/down) or y (drag left/right).
Ctrl+L	Lock Viewer to selected 3D camera or light.
Ctrl+Shift+LMB	Rotate Viewer perspective on z.
Shift+C	3D view, bottom orthographic.
Shift+X	3D view, left-side orthographic.
Shift+Z	3D view, back orthographic.
Tab	3D / 2D view toggle.
V	3D view, perspective.

Keystroke(s)	Action
X	3D view, right-side orthographic.
Z	3D view, front-side orthographic.

Bezier Draw

Keystroke(s)	Action
1, 2, 3, 4, 6, 7, 8, and 9 (numeric keypad)	Press the keys on the numeric key pad to move the selected points in the directions shown below. Press **Ctrl** and the key to nudge in 1/10 pixel increments.
C	Make selected points linear (Cusp).
Ctrl+A	Select all points.
Ctrl+Alt+LMB	Plot control point.
Ctrl+LMB	Feather selected control points.
Delete	Remove selected points.
Shift+C	Remove feather outline from selected points (Unblur).
Shift+Z	Feather outline off selected points (Blur).
X	Break tangent handles for selected points.
Z	Make tangent handles horizontal on selected points (Smooth).

Curve Editor

Keystroke(s)	Action
Alt+LMB drag	Pan inside the Curve Editor.
Alt+Shift+LMB drag	Move a single point, leaving any other selected points where they are.
Alt+MMB drag	Variable zoom: zoom in or out on the x or y axis only.
C	Change interpolation of selected control points to Cubic.
Ctrl+A	Select all curves.
Ctrl+Alt+LMB	Add a point to the current curve.
Ctrl+Alt+Shift+LMB drag	Sketch points freely on the current curve.
Ctrl+C	Copy selected keys.
Ctrl+E	Copy expressions.
Ctrl+L	Copy links.

Keystroke(s)	Action
Ctrl+LMB drag	Remove horizontal/vertical constraint on moving points.
hold down Ctrl+Shift	Hide points to make it easier to click on the selection box or transform jack.
Ctrl+V	Paste curve.
Ctrl+X	Cut selected keys.
H	Change interpolation of selected control points to Horizontal.
K	Change interpolation of selected control points to Constant.
L	Change interpolation of selected control points to Linear.
LMB	Select a single point on the curve.
LMB drag on blank space	Draw a box to select multiple points.
LMB drag on a point	Move all selected points.
LMB drag on a selection box	Resize a selection box and scale the points inside.
LMB drag on a transform jack	Move all points inside the selection box.
MMB or F	Fit selection in the window.
MMB drag	Draw a box around an area and zoom to fit that area in the Editor.
R	Change interpolation of selected control points to Catmull-Rom.
Shift+Ctrl+C	Copy selected curves.
Shift+LMB	Add or remove points to/from selection.
Shift+LMB drag	Draw box to add/remove points to/from selection.
X	Break selected control points' handles.
Z	Change interpolation of selected control points to Smooth (Bezier).

Script Editor

Keystroke(s)	Action
Ctrl+[	Step back to the previous statement.
Ctrl+]	Step forward to the next statement.
Ctrl+Backspace	Clear output pane.
Ctrl+Enter (numeric keypad)	Run the script in the Editor.

Keystroke(s)	Action
Ctrl+Return	Run the script in the Editor.
Ctrl+Shift+[	Decrease the indentation level.
Ctrl+Shift+]	Increase the indentation level.

Toolbar

Keystroke(s)	Action
MMB	Repeat the last item used from the menu.

Content Menus

Keystroke(s)	Action
Ctrl+LMB	Open the selected menu item in a floating window.

Colour Picker

Keystroke(s)	Action
drag right or left (on a slider label)	Scrub the value up or down.
Alt+drag right or left (on a slider label)	Scrub the value up or down slowly.
Alt+LMB (on a slider label)	Decrement the value by 0.001.
Alt+RMB (on a slider label)	Increment the value by 0.001.
LMB (on a slider label)	Decrement the value by 0.01.
RMB (on a slider label)	Increment the value by 0.01.
Shift+drag right or left (on a slider label)	Scrub the value up or down fast.
Shift+LMB (on a slider label)	Decrement the value by 0.1.
Shift+RMB (on a slider label)	Increment the value by 0.1.

Appendix B: Supported File Formats

This appendix lists the image and video formats recognized by Nuke. When importing and exporting image sequences remember the following:

- When you import images with a Read node (**Image > Read**), Nuke analyses the contents of the file to determine the format.The filename extension is not used to determine file format, which allows flexibility with naming conventions in a production environment.
- Regardless of format, Nuke converts all imported sequences to its native 32-bit linear RGB colorspace.
- When you render new images from Nuke (**Image > Write**), you can use a filename extension to specify format.

Supported Image Formats

The following table lists the supported image formats. The extensions listed under "Filename Extension" let you specify the image format; use these as the actual filename extensions or the prefix to indicate output format for the image sequences.

Format	Bit Depths	Read/Write	Filename Identifier
CIN	10 (log)	read and write	cin
DPX	8, 10, and 16	read and write	dpx
EXR	16 and 32[1]	read and write[2]	exr
FPI	obsolete		
GIF	8	read only	gif
HDRI	n/a	read and write	hdr
JPEG	8	read and write	jpg, jpeg[3]
Maya IFF	8 and 16	read only	iff
PNG	8 and 16	read and write	png (8-bit) png16 (16-bit)
QuickTime[4]	n/a	read and write	mov
RAW[5]	n/a	read only	n/a
SGI	8 and 16	read and write	sgi, rgb, rgba (8-bit sequences) sgi16 (for 16-bit sequences)
SoftImage® PIC	8	read and write	pic

Format	Bit Depths	Read/Write	Filename Identifier
TIFF	8, 16, and 32	read[6] and write	tif, tiff (8-bit sequences) tif16, tiff16 (16-bit sequences) ftif, ftiff (32-bit sequences)
Truevision® TARGA	8	read and write	tga, targa
Wavefront® RLA	8	read only	rla
XPM[7]	8	read and write	xpm
YUV	8	read[8] and write	yuv

1. Exr handles 16- and 32-bit float. This 16 is also called "half float" and is different from the 16-bit integer that all the other formats that support 16 use.
2. TIFF sequences read in at bit-depths up to 32, but can be written out at a maximum bit-depth of 16.
3. Adjust compression levels using the **quality** slider in the Write node's properties panel.
4. QuickTime is only supported on 32-bit Windows and Mac OS X.
5. DSLR raw data files, such as Canon .CR2 files. These are only supported via the dcraw command-line program, which you can download from the dcraw website. Bit depth and other specifications depend on the device. Some devices may not be supported.
6. If utilized, the compression schema on imported TIFF sequences must be LZW®.
7. This is the text-based format in which Nuke's interface elements are stored.
8. This format does not specify resolution, so Nuke assumes a width of 720 pixels.

Appendix C: Converting from Shake to Nuke

This appendix contains information that will assist Shake users in making the transition to Nuke. Although Nuke does provide an intuitive workflow for a broad base of compositing tasks, there will be a few cases where Shake artists may ask "How do I (*fill in the blank)* in Nuke?" The following information will help you fast-track to the answers you need.

The first part of this appendix provides practical information for Shake artists, including terms used in Nuke, the layout of the user interface, and differences in the workflow for common tasks. The second part lists the commands you use to create node trees in Shake with their counterparts in Nuke.

Terms (and Conditions)

Let's start with vocabulary differences between Shake and Nuke. The following table lists several Shake terms and their Nuke equivalents. This table does not include the names of operators you use to create a node tree; these are covered later in this appendix.

Shake Term	Nuke Equivalent	Description	More Info
Cache	Buffer	Cached or buffered image data is saved to disk so Nuke can quickly display the results without recalculating the parts of the script that have not changed.	Nuke User Manual, page 95
Command Palette	Toolbar	The palette that contains all possible nodes for creating your node tree. Shake's command palette lets you browse through the nodes according to category. In Nuke, you can do the same by rolling over the buttons on the Toolbar.	Nuke User Manual, page 33
Console	(see description)	To display commands as they are executed in Nuke, you must launch Nuke from a shell using the **-V** option. For example, Windows users would open a command line window, navigate to Nuke application directory and enter: **nuke5.1.exe -V**	
Control	Knob or control	A control that appears in the parameters of a node, such as a check box that toggles an option or a slider to change a numeric value. You can create custom "knobs" to add new controls to the Nuke properties panels.	Nuke User Manual, page 46
Globals	Settings	The place where you define the global settings for the current script, such as resolution, frame range, frames per second, and other project settings.	Nuke User Manual, page 90

Shake Term	Nuke Equivalent	Description	More Info
Interface Setting	Window Layout	Nuke's interface is highly customizable. When you find an arrangement of panes and panels that you like, your can save the window layout. Press **Ctrl/Cmd+F1** to save the layout. Press **Shift+F1** to restore. You can save up to 6 layouts (Ctrl/Cmd+F1, Ctrl/Cmd+F2, Ctrl/Cmd+F3...). To restore a layout, just press **Shift** and the function key you used to save the layout.	Nuke User Manual, page 86
Layer	Merge	The process of compositing one image with another. You know it as layering in Shake. In Nuke, it's called Merge.	
Macro	Gizmo	These are the user-defined nodes, where a series of operations are saved, tweaked, and presented as group or object that may be used again in other scripts. When defining a gizmo, you decide which "knobs" (controls) are displayed from the original group of nodes.	Nuke User Manual, page 419
Node View	Node Graph or DAG	The main workspace where you construct your node tree. In Nuke, this is sometimes referred to as "The DAG," which is old-school lingo for "The Directed Acyclic Graph."	Nuke User Manual, page 29
Noodle	Connector or pipe	The lines that connect the items in the node tree. "Shaking" a node does not disconnect it from its tree in Nuke, but pressing **Ctrl/Cmd+Shift+X** will disconnect (or extract) the node.	Nuke User Manual, page 39
Parameters	Properties panel or control panel	These are the controls that determine what a node passes to the next node in the tree. In Nuke, you can open an unlimited number of properties panels to edit the nodes in your tree—much better than Shake's limit of two parameter tabs.	Nuke User Manual, page 46
Thumbnail	Postage Stamp	The preview image on a node. In Nuke, you can show or hide a thumbnail for any node. They are refreshed automatically as changes are made to your script.	

Node Reference

These tables list the nodes as organized in the Shake user interface. Where applicable, the Nuke equivalent is listed with notes that describe any differences and conditions that may be useful to the compositing artist.

Image Nodes

Shake Node	Nuke Equivalent	Description	More Info
Checker	Image > Checkerboard		
Color	Image > Constant		
ColorWheel			
FileIn	Image > Read		Nuke User Manual, page 97
FileOut	Image > Write		Nuke User Manual, page 352
Grad			
QuickPaint	Draw > Paint		
QuickShape	Draw > Bezier		
Ramp	Draw > Ramp		
Rand			
RGrad	Draw > Radial		
RotoShape	Draw > Bezier		
Text	Draw > Text		

Color Nodes

Shake Node	Nuke Equivalent	Description	More Info
Add	Color > Math > Add		Nuke User Manual, page 154
AdjustHSV	Color > HSVTool		Nuke User Manual, page 143
Brightness	Color > Multiply		Nuke User Manual, page 155
Clamp	Color > Math > Clamp		Nuke User Manual, page 153
ColorCorrect	Color > ColorCorrect		Nuke User Manual, page 141
ColorMatch			
ColorReplace			
ColorSpace	Color > Colorspace		Nuke User Manual, page 157
ColorX	Color > Math > Expression		Nuke User Manual, page 155
Compress			
ContrastLum			
ContrastRGB	Color > RolloffContrast		
Expand			
Fade	Color > Multiply		Nuke User Manual, page 155
Gamma	Color > Math > Gamma		
HueCurves	Color > HueCorrect		Nuke User Manual, page 146
Invert	Color > Math > Invert		
LogLin	Color > Log2Lin		Nuke User Manual, page 156
Lookup	Color > ColorLookup		Nuke User Manual, page 141
LookupFile	Color > ColorLookup		Nuke User Manual, page 141

Shake Node	Nuke Equivalent	Description	More Info
LookupHLS			
LookupHSV			
MDiv	Merge > Unpremult		
MMult	Merge > Premult		
Monochrome			
Mult			
Reorder	Channel > Shuffle		Nuke User Manual, page 123
Saturation	Color > Saturation		Nuke User Manual, page 147
Set			
SetAlpha			
SetBGColor			
Solarize			
Threshold			
Truelight	Color > Truelight		
VideoSafe			

Filter Nodes

Shake Node	Nuke Equivalent	Description	More Info
ApplyFilter			
Blur	Filter > Blur		
Convolve	Filter > Convolve		
Defocus	Filter > Defocus		
DilateErode	Filter > Erode		
EdgeDetect	Filter > EdgeDetect		
Emboss	Filter > Emboss		
FilmGrain	Draw > ScannedGrain		Nuke User Manual, page 150
IBlur			
IDefocus			
IRBlur			
IDilateErode			
ISharpen			
Median	Filter > Median		
PercentBlur	Filter > Blur		
Pixelize			
RBlur	Filter > Godrays		
Sharpen	Filter > Sharpen		
Zblur	Filter > ZBlur		
ZDefocus			

Key Nodes

Shake Node	Nuke Equivalent	Description	More Info
ChromaKey	Keyer > HueKeyer		
DepthKey			
DepthSlice	Filter > ZSlice		
LumaKey	Keyer > Keyer		
Primatte	Keyer > Primatte		Nuke User Manual, page 187
SpillSuppress			
Keylight	Keyer > Keylight OFX plug-in (optional)		

Layer Nodes

Shake Node	Nuke Equivalent	Description	More Info
AddMix	Merge > AddMix		
AddText	Draw > Text	May be added "in-line" to composite text over an input without the use of a layer (merge) node. In Shake, this was added because the Image > Text node did not have an input. In Nuke, the Draw > Text node may be used in-line or layered over another image with a Merge operation.	
Atop	Merge > Merge		
Common			
Constraint			
Copy	Channel > Copy		Nuke User Manual, page 125
IAdd	Merge > Merge	Set operator = plus	
IDiv	Merge > Merge	Set operator = divide	
IMult	Merge > Merge	Set operator = multiply	
Inside	Merge > Merge	Set operator = in	
Interlace			
ISub	Merge > Merge	Set operator = minus	
ISubA	Merge > Merge	Set operator = difference	
KeyMix	Merge > Keymix		
LayerX			
Max	Merge > Merge	Set operator = max	
Min	Merge > Merge	Set operator = min	
Mix	Merge > Dissolve		
MultiLayer			
Outside	Merge > Merge	Set operator = out	
Over	Merge > Merge	Set operator = over	
Screen	Merge > Merge	Set operator = screen	
SwitchMatte			

Shake Node	Nuke Equivalent	Description	More Info
Under	Merge > Merge	Set operator = under	
Xor	Merge > Merge	Set operator = xor	
ZCompose	Merge > ZMerge		

Transform Nodes

Shake Node	Nuke Equivalent	Description	More Info

Warp Nodes

Shake Node	Nuke Equivalent	Description	More Info

Other Nodes

Shake Node	Nuke Equivalent	Description	More Info
AddBorders			
AddShadow			
Bytes			
Cache			
DeInterlace			
DropShadow			
Field			
Histogram			
PixelAnalyzer	Image > CurveTool		Nuke User Manual, page 273
PlotScanline			
Select			
SwapFields			
Tile			
TimeX			
TLCalibrate			
Transition			

Appendix D: Third Party Licenses

This appendix lists third party libraries used in Nuke, along with their licenses.

Library	Description	License
Boost	Source code function / template library	Boost Software License - Version 1.0 - August 17th, 2003 Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following: The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	Description	License
Expat	XML parser	Copyright © 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper Copyright © 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Autodesk FBX	File format support	This software contains Autodesk® FBX® code developed by Autodesk, Inc. Copyright 2008 Autodesk, Inc. All rights, reserved. Such code is provided "as is" and Autodesk, Inc. disclaims any and all warranties, whether express or implied, including without limitation the implied warranties of merchantability, fitness for a particular purpose or non-infringement of third party rights. In no event shall Autodesk, Inc. be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of such code.
FreeType	Font support	Portions of this software are copyright © 2008 The FreeType Project (www.freetype.org). All rights reserved.

Library	Description	License
GLEW	OpenGL support	The OpenGL Extension Wrangler Library Copyright (C) 2002-2008, Milan Ikits <milan ikits□ieee org> Copyright (C) 2002-2008, Marcelo E. Magallon <mmagallo□debian org> Copyright (C) 2002, Lev Povalahev All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * The name of the author may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	Description	License
FTGL	OpenGL support	FTGL - OpenGL font library Copyright © 2001-2004 Henry Maddocks ftgl@opengl.geek.nz Copyright © 2008 Sam Hocevar sam@zoy.org Copyright © 2008 Sean Morrison learner@brlcad.org Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
IJG JPEG	File format support	This software is based in part on the work of the Independent JPEG Group.

Library	Description	License
Lib Tiff	File Format Support	Copyright © 1988-1997 Sam Leffler Copyright © 1991-1997 Silicon Graphics, Inc. Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics. THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Library	Description	License
OFX	Plug-in API	Copyright © 2003-2004, The Foundry Visionmongers Ltd. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name The Foundry Visionmongers Ltd, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	Description	License
OpenEXR	File format support	Copyright © 2002, Industrial Light & Magic, a division of Lucas Digital Ltd. LLC All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Industrial Light & Magic nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Python	Source code language	Copyright © 2001, 2002, 2003, 2004 Python Software Foundation; All Rights Reserved.

Library	Description	License
TCL	Source code language	This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files. The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply. IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS. GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 © (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 © (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

Library	Description	License
VXL	Computer vision	Copyright © 2000-2003 TargetJr Consortium GE Corporate Research and Development (GE CRD) 1 Research Circle Niskayuna, NY 12309 All Rights Reserved Reproduction rights limited as described below. Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notice and this permission notice appear in all copies of the software and related documentation, (ii) the name TargetJr Consortium (represented by GE CRD), may not be used in any advertising or publicity relating to the software without the specific, prior written permission of GE CRD, and (iii) any modifications are clearly marked and summarized in a change history log. THE SOFTWARE IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL THE TARGETJR CONSORTIUM BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR ON ANY THEORY OF LIABILITY ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
SparseLib++	Source code library	SparseLib++ : Sparse Matrix Library National Institute of Standards and Technology University of Notre Dame Authors: R. Pozo, K. Remington, A. Lumsdaine NOTICE Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted provided that the above notice appear in all copies and supporting documentation. Neither the Institutions (National Institute of Standards and Technology, University of Notre Dame) nor the Authors make any representations about the suitability of this software for any purpose. This software is provided "as is" without expressed or implied warranty.
InventorMath	Maths library	Open Inventor code is copyright SGI. All rights reserved.

Appendix E: End User Licensing Agreement

This END USER SOFTWARE LICENSE AGREEMENT (this "Agreement") is made by and between The Foundry Visionmongers Ltd., a company registered in England and Wales, ("The Foundry"), and you, as either an individual or a single entity ("Licensee").

In consideration of the mutual covenants contained herein and for other good and valuable consideration (the receipt and sufficiency of which is acknowledged by each party hereto) the parties agree as follows:

SECTION 1. GRANT OF LICENSE.

Subject to the limitations of Section 2, The Foundry hereby grants to Licensee a limited, non-transferable and non-exclusive license to install and use a machine readable, object code version of this software program (the "Software") and the accompanying user guide and other documentation (collectively, the "Documentation") solely for Licensee's own internal business purposes (collectively, the "License"); provided, however, Licensee's right to install and use the Software and the Documentation is limited to those rights expressly set out in this Agreement.

SECTION 2. RESTRICTIONS ON USE.

Licensee is authorized to use the Software in machine readable, object code form only, and Licensee shall not: (a) assign, sublicense, transfer, pledge, lease, rent, share or export the Software, the Documentation or Licensee's rights hereunder; (b) alter or circumvent the copy protection mechanisms in the Software or reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software; (c) modify, adapt, translate or create derivative works based on the Software or Documentation; (d) use, or allow the use of, the Software or Documentation on any project other than a project produced by Licensee (an "Authorized Project"); (e) allow or permit anyone (other than Licensee and Licensee's authorized employees to the extent they are working on an Authorized Project) to use or have access to the Software or Documentation; (f) copy or install the Software or Documentation other than as expressly provided for herein; or (g) take any action, or fail to take action, that could adversely affect the trademarks, service marks, patents, trade secrets, copyrights or other intellectual property rights of The Foundry or any third party with intellectual property rights in the Software (each, a "Third Party Licensor"). Furthermore, for purposes of this Section 2, the term "Software" shall include any derivatives of the Software.

Licensee shall install and use only a single copy of the Software on one computer, unless the Software is installed in a "floating license" environment, in which case Licensee may install the Software on more than one computer; provided, however, Licensee shall not at any one time use more copies of the Software than the total number of valid Software licenses purchased by Licensee.

Furthermore, the Software can be licensed on an "interactive" or "non-interactive" basis. Licensee shall be authorized to use a non-interactive version of the Software for rendering purposes only (i.e., on a CPU, without a user, in a non-interactive capacity) and shall not use such Software on workstations or otherwise in a user-interactive capacity. Licensee shall be authorized to use an interactive version of the Software for both interactive and non-interactive rendering purposes.

Finally, if the Software is a "Personal Learning Edition", Licensee may use it only for the purpose of training and instruction, and for no other purpose. PLE versions of the Software may not be used for commercial, professional or for-profit purposes.

SECTION 3. BACK-UP COPY.

Notwithstanding Section 2, Licensee may store one copy of the Software and Documentation off-line and off-site in a secured location owned or leased by Licensee in order to provide a back-up in the event of destruction by fire, flood, acts of war, acts of nature, vandalism or other incident. In no event may Licensee use the back-up copy of the Software or Documentation to circumvent the usage or other limitations set forth in this Agreement.

SECTION 4. OWNERSHIP.

Licensee acknowledges that the Software and Documentation and all intellectual property rights relating thereto are and shall remain the sole property of The Foundry and the Third Party Licensors. Licensee shall not remove, or allow the removal of, any copyright or other proprietary rights notice included in and on the Software or Documentation or take any other action that could adversely affect the property rights of The Foundry or any Third Party Licensor. To the extent that Licensee is authorized to make copies of the Software or Documentation under this Agreement, Licensee shall reproduce in and on all such copies any copyright and/or other proprietary rights notices provided in and on the materials supplied by The Foundry hereunder. Nothing in this Agreement shall be deemed to give Licensee any rights in the trademarks, service marks, patents, trade secrets, copyrights or other intellectual property rights of The Foundry or any Third Party Licensor, and Licensee shall be strictly prohibited from using the name, trademarks or service marks of The Foundry or any Third Party Licensor in Licensee's promotion or publicity without The Foundry's express written approval.

SECTION 5. LICENSE FEE.

Licensee understands that the benefits granted to Licensee hereunder are contingent upon Licensee's payment in full of the license fee payable in connection herewith (the "License Fee").

SECTION 6. UPGRADES/ENHANCEMENTS.

As long as Licensee pays the applicable fee for annual support, upgrades and updates ("Annual Upgrade and Support Program"), The Foundry or its authorized reseller will provide Licensee with access to upgrades and updates, if any, made available by The Foundry on the terms and conditions set forth in this Agreement, unless such upgrade or update contains a separate license. The term shall begin upon the date of purchase and continue for 12 months, unless otherwise expressly agreed to in writing by The Foundry. Before requesting technical support, purchasers of the Annual Upgrade and Support Program must send an email to support@thefoundry.co.uk in order to register the names of up to 2 employees who shall then become the technical representatives of the purchaser. The Foundry will endeavour to provide news of upgrades and technical support to the technical representatives; however, The Foundry cannot guarantee resolution or the results of any assistance that may be provided. The Foundry's technical support personnel can be contacted by phone Monday through Friday (excluding holidays) during normal business hours 0930-1800, and by email at support@thefoundry.co.uk. Subsequent renewals of the Annual Upgrade and Support Program will be charged at the rate shown in The Foundry's current price list.

SECTION 7. TAXES AND DUTIES.

Licensee agrees to pay, and indemnify The Foundry from claims for, any local, state or national tax (exclusive of taxes based on net income), duty, tariff or other impost related to or arising from the transaction contemplated by this Agreement.

SECTION 8. LIMITED WARRANTY.

The Foundry warrants that, for a period of ninety (90) days after delivery of the Software: (a) the machine readable electronic files constituting the Software and Documentation shall be free from errors that may arise from the electronic file transfer from The Foundry and/or its authorized reseller to Licensee; and (b) to the best of The Foundry's knowledge, Licensee's use of the Software in accordance with the Documentation will not, in and of itself, infringe any third party's copyright, patent or other intellectual property rights. Except as warranted, the Software and Documentation is being provided "as is." THE FOREGOING LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, AND THE FOUNDRY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES OR CONDITIONS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDLESS OF WHETHER THE FOUNDRY KNOWS OR HAS REASON TO KNOW OF LICENSEE'S PARTICULAR NEEDS. The Foundry does not warrant that the Software or Documentation will meet Licensee's requirements or that Licensee's use of the Software will be uninterrupted or error free. No employee or agent of The Foundry is authorized to modify this limited warranty, nor to make additional warranties. No action for any breach of the above limited warranty may be commenced more than one (1) year after Licensee's initial receipt of the Software. To the extent any implied warranties may not be disclaimed under applicable law, then ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO NINETY (90) DAYS AFTER DELIVERY OF THE SOFTWARE TO LICENSEE.

SECTION 9. LIMITED REMEDY.

The exclusive remedy available to the Licensee in the event of a breach of the foregoing limited warranty, TO THE EXCLUSION OF ALL OTHER REMEDIES, is for Licensee to destroy all copies of the Software, send The Foundry a written certification of such destruction and, upon The Foundry's receipt of such certification, The Foundry will make a replacement copy of the Software available to Licensee.

SECTION 10. INDEMNIFICATION.

Licensee agrees to indemnify, hold harmless and defend The Foundry and The Foundry's affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives (collectively, the "Released Parties") from all claims, defense costs (including, but not limited to, attorneys' fees), judgments, settlements and other expenses arising from or connected with the operation of Licensee's business or Licensee's possession or use of the Software or Documentation.

SECTION 11. LIMITED LIABILITY.

In no event shall the Released Parties' cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands or actions arising out of or relating to this Agreement (or the Software or Documentation contemplated herein) exceed the License Fee paid to The Foundry or its authorized reseller for use of the Software. Furthermore, IN NO EVENT SHALL THE RELEASED PARTIES BE LIABLE TO LICENSEE UNDER ANY THEORY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, EXEMPLARY OR CONSEQUENTIAL DAMAGES

(INCLUDING DAMAGES FOR LOSS OF BUSINESS OR LOSS OF PROFITS) OR THE COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, REGARDLESS OF WHETHER THE RELEASED PARTIES KNOW OR HAVE REASON TO KNOW OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF WHETHER ANY REMEDY SET FORTH HEREIN FAILS OF ITS ESSENTIAL PURPOSE. No action arising out of or related to this Agreement, regardless of form, may be brought by Licensee more than one (1) year after Licensee's initial receipt of the Software; provided, however, to the extent such one (1) year limit may not be valid under applicable law, then such period shall limited to the shortest period allowed by law.

SECTION 12. TERM; TERMINATION.

This Agreement is effective upon Licensee's acceptance of the terms hereof (by clicking on the "Accept" button on installation) and Licensee's payment of the License Fee, and the Agreement will remain in effect until termination. If Licensee breaches this Agreement, The Foundry may terminate the License granted hereunder by notice to Licensee. In the event the License is terminated, Licensee will either return to The Foundry all copies of the Software and Documentation in Licensee's possession or, if The Foundry directs in writing, destroy all such copies. In the later case, if requested by The Foundry, Licensee shall provide The Foundry with a certificate signed by an officer of Licensee confirming that the foregoing destruction has been completed.

SECTION 13. CONFIDENTIALITY.

Licensee agrees that the Software and Documentation are proprietary and confidential information of The Foundry and that all such information and any communications relating thereto (collectively, "Confidential Information") are confidential and a fundamental and important trade secret of The Foundry. Licensee shall disclose Confidential Information only to Licensee's employees who are working on an Authorized Project and have a "need-to-know" such Confidential Information, and shall advise any recipients of Confidential Information that it is to be used only as authorized in this Agreement. Licensee shall not disclose Confidential Information or otherwise make any Confidential Information available to any other of Licensee's employees or to any third parties without the express written consent of The Foundry. Licensee agrees to segregate, to the extent it can be reasonably done, the Confidential Information from the confidential information and materials of others in order to prevent commingling. Licensee shall take reasonable security measures, which such measures shall be at least as great as the measures Licensee uses to keep Licensee's own confidential information secure (but in any case using no less than a reasonable degree of care), to hold the Software, Documentation and any other Confidential Information in strict confidence and safe custody. The Foundry may request, in which case Licensee agrees to comply with, certain reasonable security measures as part of the use of the Software and Documentation. Licensee acknowledges that monetary damages may not be a sufficient remedy for unauthorized disclosure of Confidential Information, and that The Foundry shall be entitled, without waiving any other rights or remedies, to such injunctive or equitable relief as may be deemed proper by a court of competent jurisdiction.

SECTION 14. INSPECTION.

Licensee shall advise The Foundry on demand of all locations where the Software or Documentation is used or stored. Licensee shall permit The Foundry or its authorized agents to inspect all such locations during normal business hours and on reasonable advance notice.

SECTION 15. NONSOLICITATION.

Licensee agrees not to solicit for employment or retention, and not to employ or retain, any of The Foundry's current or future employees who were or are involved in the development and/or creation of the Software.

SECTION 16. U.S. GOVERNMENT LICENSE RIGHTS.

The Software, Documentation and/or data delivered hereunder are subject to the terms of this Agreement and in no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication or disclosure by the U.S. Government is subject to the applicable restrictions of: (i) FAR §52.227-14 ALTS I, II and III (June 1987); (ii) FAR §52.227-19 (June 1987); (iii) FAR §12.211 and 12.212; and/or (iv) DFARS §227.7202-1(a) and DFARS §227.7202-3.

The Software is the subject of the following notices:

- Copyright (c) 2009 The Foundry Visionmongers Ltd. All Rights Reserved.
- Unpublished-rights reserved under the Copyright Laws of the United Kingdom.

SECTION 17. SURVIVAL.

Sections 2, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18 and 19 shall survive any termination or expiration of this Agreement.

SECTION 18. IMPORT/EXPORT CONTROLS.

To the extent that any Software made available hereunder is subject to restrictions upon export and/or reexport from the United States, Licensee agrees to comply with, and not act or fail to act in any way that would violate, the applicable international, national, state, regional and local laws and regulations, including, without limitation, the United States Foreign Corrupt Practices Act, the Export Administration Act and the Export Administration Regulations, as amended or otherwise modified from time to time, and neither The Foundry nor Licensee shall be required under this Agreement to act or fail to act in any way which it believes in good faith will violate any such laws or regulations.

SECTION 19. MISCELLANEOUS.

This Agreement is the exclusive agreement between the parties concerning the subject matter hereof and supersedes any and all prior oral or written agreements, negotiations, or other dealings between the parties concerning such subject. This Agreement may be modified only by a written instrument signed by both parties. If any action is brought by either party to this Agreement against the other party regarding the subject matter hereof, the prevailing party shall be entitled to recover, in addition to any other relief granted, reasonable attorneys' fees and expenses of litigation. Should any term of this Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms of this Agreement. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches. This Agreement shall be governed by, and construed in accordance with English law.

Copyright (c) 2009 The Foundry Visionmongers Ltd. All Rights Reserved. Do not duplicate.

Index

Numerics

2.5D transformations 169
2D transformations 159
3D 513
3D geometry 533
3D objects
  normals 315
3D scenes, lighting 309
3D viewers 513
3D workspace 277

A

Add nodes 154
adding
  a black edge to the bounding box 116
  Axis objects to 3D scenes 289
  Card objects to 3D scenes 283
  Cube objects to 3D scenes 286
  mathematical functions to expressions 359
  motion blur 173
  nodes 35
  OBJ-type objects to 3D scenes 288
  Sphere objects to 3D scenes 287
  viewers 66
AdjBBox nodes 114
adjusting
  contrast 140
  gain 140
  gamma 140
  HSV 144
  hue 143
  offset 140
  saturation 143
  value 143
anaglyph images, converting into 342
Anaglyph nodes 342
analysing frame sequences 273
analysing pixels 273
animating parameters 56
AppendClip nodes 245
applying
  image formats 110
  motion blur 243
  tracking data to clips 183
Axis nodes 289
  adding to scenes 289

B

Backdrop nodes 102
BasicMaterial nodes 541
Bezier nodes 498
Bicubic nodes 283
black points, sampling 139
BlackOutside nodes 116
bounding box
  adding a black edge to 116
  copying from an image 115
  resizing 114

C

calling, channels 118
cameras
  dollying 281
  editing lens characteristics of 316
  resetting 281
  rolling 281
Card nodes 283
  adding to 3D scenes 283
CCorrect nodes 141
changing
  convergence 343
channel count 102
channel values
  applying expressions to 155
  clamping 153
  inverting 154
  multiplying 155
  offsetting 154
channels 117
  applying mathematical operations to 153
  calling 118
  creating 117
  deleting 122
  displaying in a viewer 72
  renaming 122
  separating 51
  swapping 123
  tracing 122
CIN format 557
Cineon conversions 156
Clamp nodes 153
clamping channel values 153
Clean BG Noise 190
Clean FG noise 191
clips
  applying tracking data to 183
  corner-pinning 184
  cutting frames from 244
  dissolving between 245
  editing 244
  fading to black 245
  retiming 233
  slipping 244
  splicing 245
  stabilizing 185
  warping 241
cloning nodes 38
Colorspace nodes 157
colour curves 141
colour picker 52
colour pickers 51
colour sliders 52
colour space conversions 156
colour wheel 52
configuring Nuke 401
connecting
  nodes 39
  viewers 67
contrast, adjusting 140
control panels 52
  displaying 49
convergence
  changing 343
CopyBBox nodes 115
copying
  a bounding box from an image 115
  a rectangle from an image 135
  nodes 37
CopyRectangle nodes 135
CornerPin2D nodes 184
  applying tracking data to 185
corner-pinning clips 184
creating
  bg reflections on fg elements 261
  channels 117
  effects 261
  formats with the Reformat node 109
  layers 117
  star filter effects 263
Crop nodes 113
cropping elements 113
CrosstalkGeo nodes 302
Cube nodes 286
  adding to 3D scenes 286
Cubic filtering algorithm 161
curve editor 57
curves, editing tracks with 180
CurveTool nodes 273
custom plug-ins 419, 430
customer support 17
customising
  nodes 40
  Nuke 401
cutting
  a clip's frames 244
  nodes 37

D

defining
  common image formats 419
  common menu options 414
  common preferences 431
  tonal range 138
Degrain tools, Primatte 203
delete preferences 432
deleting
  channels 122
  image formats 110
  keyframes 66
  layers 122

nodes 39
DirectLight nodes 310
disabling nodes 39
DisplaceGeo nodes 306
display gain 80
display gamma 80
displaying node parameters 49
displaying script information 102
dissolving between clips 245
distorting time 233
Dither nodes 149
Dot nodes 40
DPX format 557
dual monitors 86

E

editing
camera lens characteristics 316
clips 244
image formats 110
nodes 37
parameters 49
tracks with curves 180
tracks with overlays 179
elements
cropping 113
rotating in 2.5D 172
rotating in 2D 166
scaling in 2.5D 172
scaling in 2D 167
skewing in 2.5D 172
skewing in 2D 168
translating in 2.5D 171
translating in 2D 165
environment light 312
Environment nodes 313
expressions 357
adding mathematical functions to 359
applying to channel values 155
applying tracking data with 183
linking 357
syntax of 357
Expressions nodes 155
EXPTool nodes 464
EXR format 557

F

FFT 20
file formats, supported 557
file name conventions 355
FilmGrain nodes 149
filtering algorithms 169
selecting 160
fine tuning sliders, Primatte 208
flipbooking sequences 350, 466
formats
CIN 557
creating with the reformat node 109
DPX 557
EXR 557
FPI 557
GIF 557
HDRI 557
IFF 557
JPEG 557
MOV 557
PIC 557
PNG 557
RAW 557
RLA 558
SGI 557
Targa 558
TIFF 558
XPM 558
YUV 558
Foundry Flexlm Tools 20
four-point tracking 472
FPI format 557
Frame Blend nodes 235
frame ranges 100
FrameCycler 351
FrameRange nodes 244
frames
cutting from clips 244
interpolating between for temporal effects 234
full size formats 91
full size mode 91
full-size formats 91
full-size mode 92

G

gain
adjusting 140
displayed in a viewer 80
gamma
adjusting 140
displayed in a viewer 80
generating, tracks 178
GIF format 557
gizmos 419
Glint nodes 263
global frame range 242
Grade nodes 139
grain
matching 149
practical 150
synthetic 149
Grain nodes 149
GridWarp nodes 247, 256, 258

H

HDRI format 557
help, online 16
histograms 139
Host ID number 20
hot keys 546
HSV, adjusting 144
HSVTool nodes 144
hue, adjusting 143
HueCorrect nodes 146

I

IBK 492
IFF format 557
iitter, removing from tracks 180
image formats 109
applying 110
defining common 419
deleting 110
displaying in a viewer 74
editing 110
supported 557
Image-Based Keyer 492
Impulse filtering algorithm 161
indicators
on nodes 41
input channels, selecting 122
input fields 50
interface preferences 89, 367
Invert nodes 154
inverting channel values 154

J

JoinViews nodes 337
JPEG format 557

K

keyboard shortcuts 546
Keyer nodes 504
keyframes
deleting 66
setting 56
keying 187, 485
keying video 504
Keys filtering algorithm 162

L

launching Nuke 25, 440
layering images 126
layers 117
creating 117
deleting 122
license keys 20
licenses
floating 20
node-locked 20
obtaining 20
licensing 25
Light nodes 314
lighting 540
LightWrap nodes 261
linking expressions 357
lin-to-log conversions 156
lmhostid number 20
Log2Lin nodes 157

LogGeo nodes 304
Lookup nodes 142
lookup tables
  altering for a script 433
  default settings 436
LookupGeo nodes 302
LUTs
  altering for a script 433
  default settings 436

M

macros. See gizmos.
masking colour corrections 148
masking effects 465
masks, selecting 120
matchmoving 471, 481
mathematical operations 153
menu options, defining 414
Merge nodes 126
MergeGeo nodes 290
merging images 126
Mitchell filtering algorithm 163
MixViews nodes 334
monitors, dual 86
morphing 247, 257
motion blur, adding 173
motion blur, applying 243
MotionBlur3D nodes 320
MOV format 557
Multiply nodes 155
multiplying channel values 155

N

naming, rendered images 355
node count 102
nodes
  adding 35
  cloning 38
  connecting 39
  copying 37
  customising 40
  cutting 37
  decloning 38
  deleting 39
  disabling 39
  editing 37
  pasting 38
  previewing output from 349
  selecting 35
Notch filtering algorithm 164

O

objects
  Axis type 289
  Card type 283
  controlling display of in viewer 296
  Cube type 286
  OBJ type 288
  skewing 299
  Sphere type 287
offset, adjusting 140
offsetting
  channel values 154
  tracks 179
OFlow nodes 237
one-point tracking 472
OneView nodes 337
online help 16
overlays, for tracks 179

P

Paint nodes 220
panning
  in the node graph 44
  in viewers 68
parameters
  animating 56
  displaying for nodes 49
  editing 49
Parzen filtering algorithm 164
pasting, nodes 38
pausing, viewers 71
Phong nodes 293
PIC format 557
pivot points 300
pixel analyzer 273
pixel aspect ratio 78, 109
pixel values 74
playback speed of clips 242
PNG format 557
point light 310
Point nodes 310
Position nodes 165
postage stamps, generating 109
practical grain 150
preface 16
preferences 367
  defining common 431
  interface 89
  reset 432
preview
  flipbook 349
  region of interest (ROI) 349
previewing outputs 349
Primatte 187
  algorithm 199, 210
  Auto-Compute 188
  Degrain tools 203
  RT 218
  RT Algorithm 200
  RT+ 217
  RT+ Algorithm 199
  Select BG Color 188
Primatte nodes 486
ProcGeo nodes 307
Project3D nodes 295
proxy files 94
proxy formats 92
proxy mode 75, 92, 460
proxy scale 92
PyQt 398
Python 381
Python statements 385

Q

QuickTime format 557

R

RadialDistort nodes 308
Ramp nodes 449
RAW format 557
ReadGeo nodes 288
redoing actions 66
Reformat nodes 109, 460
reformatting elements 109
region of interest 79, 349
removing
  jitter from tracks 180
  spill 193, 194
renaming channels 122
renaming nodes 37
render
  farms 356
  format 351
  resolution 351
rendering 467
  3D scenes 279
  a sequence to flipbook 350
  from single Write nodes 353
  scripts 349, 351
repeatable sampling, Primatte 194
resizing the bounding box 114
resolution 109
  define custom 352
Retime nodes 233
retiming
  clips 233
  with OFlow 237
Rifmen filtering algorithm 163
RLA format 558
ROI (region of interest) 79, 349
rolling, cameras 281
rotating
  cameras 281
  elements in 2.5D 172
  elements in 2D 166
rotoscoping 498

S

sampling
  black points 139
  white points 139
Saturation nodes 147
saturation, adjusting 143
scaling elements
  in 2.5D 172
  in 2D 167
ScanlineRender nodes 279, 515
Scene nodes 278, 523
scenes

adding Axis objects to 289
adding Cube objects to 286
adding OBJ-type objects to 288
adding Sphere objects to 287
script editor 381
script information
displaying 102
scripting languages, Python 381
scripts 83
global frame range 242
playback speed 242
rendering 349, 351
searching
for nodes 42
selecting
filtering algorithms 160
masks 120
nodes 35
setting keyframes 56
setting up
scripts 90
tracks 175
views 329
SGI format 557
ShuffleViews nodes 341
Simon filtering algorithm 162
skewing
elements in 2.5D 172
elements in 2D 168
objects 299
sliders 51
slipping clips 244
smoothing tracks 180
spacebar 88
Sphere nodes 287
Sphere objects
adjusting geometric resolution of 288
spill
removing 193, 194
replacement methods 196, 209
suppressing 146
splicing clips 245
SplineWarp nodes 252, 256, 259
SplitAndJoin 337
Spotlight nodes 311, 542
stabilising 471, 478
Stabilize2D nodes
applying tracking data to 186
stabilizing clips with 186
stabilizing clips 185
stereoscopic images
reading 331
rendering 347
stereoscopic projects 329
displaying views 333
editing views 335
setting up views 329
StickyNote nodes 105
supported file formats 557
swapping channels 123
synchronising viewer playback 71
synthetic grain 149
system requirements 19

T

Targa format 558
TCL 357, 419, 430
using in expressions 359
temporal operations 233
Text nodes 265
third party licenses 570
three-point tracking 472
TiCkLe 357, 419, 430
TIFF format 558
Time Cut nodes 244
time distortion 233
time-based operations 233
TimeBlend nodes 236
TimeBlur nodes 243
timeline controls 69
TimeOffset nodes 244
tonal range
defining 138
defining with a histogram 139
tracing channels 122
Tracker nodes 175, 176, 471
tracking 471
per vertex 183
tracks
editing 179
generating 178
offsetting 179
pattern area for 177
positioning anchors for 177
search area for 177
smoothing 180
Transform nodes 165, 166, 167, 168
Transform2D nodes 166
Transform3D nodes 169, 171
transformation handles 170
transformations 159
choosing order of operations for 299
in 2.5D 169
in 2D 159
moving pivot for objects 300
skewing objects 299
TransformGeo nodes 298
translating
a Cube object's sides 286
cameras 281
elements in 2.5D 171
elements in 2D 165
Trilinear nodes 309
two-point tracking 472

U

undoing actions 66
UVProject nodes 294

V

value, adjusting 143
vertex tracking 183
viewer lookup tables 80
viewer LUTs 80
viewer playback, synchronising 71
viewers 66, 456
adding 66
changing colourspace 157
comparison wipes 81
configuring display of 3D objects in 296
connecting 67
display composite models 81
displaying views 333
panning 68
pausing 71
using the controls 69
zooming 68
views, setting up for project 329

W

warping 247
warping clips 241
white points, sampling 139
window layouts 86, 443
wipe control in viewers 81
Write nodes 352
wxPython 396

X

XPM format 558

Y

YUV format 558

Z

zooming
in the node graph 46
in viewers 68